Nonlinear Filtering with Particle Filters

Data assimilation on convective scale

Mylène Haslehner

München 2014

Nonlinear Filtering with Particle Filters

Data assimilation on convective scale

Mylène Haslehner

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Mylène Haslehner
aus Salzburg

München, den 5. Juni 2014

Erstgutachter: Prof. Dr. G. C. Craig und Dr. T. Janjic

Zweitgutachter: Prof. Dr. W.-C. Müller

Tag der mündlichen Prüfung: 21.7.2014

Contents

Zusammenfassung **iii**

Abstract **v**

1 Introduction **1**

2 Basic Principles of stochastic filtering **7**

2.1 Problem setting 7

2.2 The limits of linear filtering 7

2.2.1 Kalman filter 8

2.2.2 Ensemble Kalman Filter 10

2.3 Nonlinear filtering 13

2.3.1 Sequential Importance Sampling (SIS) 14

2.3.2 Importance density 15

2.3.3 Problem of degeneracy 17

2.3.4 More efficient particle filters 18

3 Model hierarchy and methodology **23**

3.1 Model hierarchy 24

3.1.1 The Stochastic Cloud model 24

3.1.2 The Shallow Water model 24

3.2 Efficient Particle filter with weight equalization 28

3.2.1 Gaussian model errors 28

3.2.2 Weight equalization and resampling 30

4 Testing the Efficient Particle Filter on the stochastic cloud model **35**

4.1 Efficient particle filter without weight equalization 35

4.1.1 Deriving the weights for a non-Gaussian model 35

4.1.2 Consequences on the filter 43

4.2 Test settings 48

4.3 Results 52

4.3.1 Cloud variability and observation error density 53

4.3.2 Conclusion 58

5 Testing the modified Shallow water model 61
5.1 Adapting the stochastic perturbation design to the filter 62
5.2 Test settings and parameters 64
5.3 Results 67
5.3.1 Preliminary tests 68
5.3.2 Scarce observations 76
5.3.3 Impact of the nudging matrix 85
5.3.4 Conclusion 101
5.4 Applicability conditions for the filter 103
5.4.1 A complex selection mechanism 104
5.4.2 Variance of the densities 115
5.4.3 Indication criterium for the success of resampling 123
5.4.4 Conclusion 127

6 Conclusion 129

Appendix 131

Bibliography 133

Acknowledgements 139

Zusammenfassung

In dieser Arbeit werden nichtlineare Datenassimilationsmethoden an einer Hierarchie von idealisierten, rechengünstigen Wolken-Modellen auf konvektiver Skala getestet. Konvektive Prozesse wie Gewitter unterliegen sehr schnellen und augenscheinlich stochastischen Veränderungen. Deshalb sind sie mit bisherigen Methoden, welche auf Linearisierungen oder Gauss'schen Fehlervermutungen basieren, schwer vorherzusagen. Ziel dieser Dissertation war es, neuere Methoden der Datenassimilation, sogenannte *Particle Filters* zu testen, die speziell für nichtlineare Modelle konzipiert wurden. Für diese Studie wurden zwei Test-Modelle [58],[59] ausgewählt, die sich aufgrund ihrer extremen Nichtlinearität und Nicht-Gaussheit speziell für die Erkundung des Particle Filter eignen. Mit Hilfe der Ergebnisse werden potentielle Probleme sowohl der Methodik als auch der Modell-Entwürfe erörtert und erste Schlussfolgerungen in Hinblick auf zukünftige Anwendungen in realen Wettermodellen gezogen.

Das erste und einfachste Modell in der Hierarchie für Konvektion [58] ist ein eindimensionaler Poisson-ähnlicher 'birth-death' Prozess von Wolkenzahlen. Das Modell ist physikalisch unzureichend realistisch, erfüllt jedoch die minimalen Merkmalsanforderungen an einen konvektiven Cumulus-Zyklus da es *rein stochastisch*, nicht Gauss'sch und ausgeprägt nicht-linear ist.

Die unbekannte Wahrscheinlichkeits-Dichtefunktion (pdf) des Modell-Fehlers wird mathematisch hergeleitet. Die Kenntnis der Modell-pdf ermöglicht es, den Effekt der Dynamik der Zustandsvektoren ('Particles') auf die Replikations-Wahrscheinlichkeit ('Gewichte') der Particles zum Zeitpunkt der Assimilierung neuer Beobachtungen zu verdeutlichen. Die Wahrscheinlichkeit einer Zustands-*Änderung* kann die Replikation eines beobachtungsnahen Particles verhindern oder die Wiederwahl eines beobachtungsfernen Particles begünstigen. Ob beobachtungsnahe Particles trotz Nudging (des 'Heranziehens' der Particles an die Beobachtungen) die grössten Gewichte aufweisen, hängt von der Magnitude der Beobachtungsverteilung und von der Langlebigkeit der Wolken ab. Es wird somit deutlich, dass der Erfolg des Particle Filters auf dem Verhältnis zwischen Modell- und Beobachtungs-Wahrscheinlichkeitsdichte basiert. Je grösser die Beobachtungs-Dichtefunktion im Vergleich zur Übergangs-Wahrscheinlichkeit ist, desto besser schneidet der Particle Filter ab. So liefert der Test mit Gauss-verteilten Beobachtungen dementsprechend grössere zeitliche RMS (Root Mean Square) Fehler als der Test mit exponentiell-verteilten Beobachtungen. Weiters wirkt sich ein sich sehr rasch veränderndes Wolkenfeld negativ auf den Filter aus, da dieser keine Zeit mehr hat, den Beobachtungen zu folgen. Ein hemmender Faktor für den Filter ist auch die Tatsache, dass der Prozess diskret ist. Das macht ein kontinuierlich konzipiertes Annähern der Particles an die Beobachtungen schwierig und verhindert eine unverzerrte Deutung des Erfolgs dieser Methode. Das stochastische Wolkenmodell stellt somit eine extreme Testumgebung für den Particle Filter dar. Trotz der genannten Nachteile dieses Testmodells kann der Particle Filter für stationäre Wolkenfelder und exponentielle Beobachtungsverteilungen bessere Ergebnisse als der herkömmliche SIR-Filter erzielen. Ausserdem beleuchtet das stochastische Wolkenmodell die Balance zwischen Beobachtungs- und Übergangsverteilung der Particles, die für das Verständnis des Particle Filter von zentraler Bedeutung ist.

Das zweite, physikalisch realistischere eindimensionale Test-Modell [59] basiert auf den Flachwassergleichungen und repräsentiert drei miteinander gekoppelte dynamische Variablen (Windgeschwindigkeit, Wasserniveau und Regenkonzentration), welche Störungen mittels Schwerewellen weitertransportieren. Im Gegensatz zum vorherigen Modell erfährt hier ein physikalisch deterministisches Modell eine additive Störung, die räumliche Korrelationen enthält. Folgende Ergebnisse bei der Anwendung des Particle Filter wurden gefunden.

Der RMS-Fehler hängt vom Nudging-Faktor ab. Es existiert ein Nudging Parameter, der zu einem minimalen RMS-Fehler führt. Für das Flachwassermodell ist es möglich, mit dem Particle Filter einen minimalen Fehler zu erreichen, der kleiner ist als der durch alleiniges Nudging (d.h. ohne Resampling) erzielte RMS Fehler. Eine Reduzierung der Beobachtungen und der Anzahl an assimilierten Variablen führt zu einem Anstieg des RMS-Fehlers. Lediglich im Falle reduzierter Beobachtungen kann man eine bessere Leistung des Particle Filter im Vergleich zu Nudging beobachten. Dabei scheint es eine maximale Reduzierung an Beobachtungen zu geben, oberhalb derer der Particle Filter vollkommen versagt. Es wird ausserdem festgestellt, dass eine diagonal basierte Nudging-Matrix zu besseren Ergebnissen führt als eine Nudging-Matrix, die auf der Modellfehler-Covarianz-Matrix basiert.

Weiters werden die Bedingungen für ein besseres Gelingen des Particle Filter im Vergleich zu Nudging untersucht. Die Standardabweichung der Log-Dichtefunktionen in den Gewichten liefert ein numerisches Kriterium dafür, unter welcher Bedingung das Resampling dem Filter zu einem Vorteil verleiht. Eine theoretische Abschätzung der Standard-Abweichung dieser Dichte-Exponenten wird vorgenommen und mit numerischen Daten verglichen. Theorie und Experiment stimmen im Fall der Assimilierung einer Variable überein.

Der Particle Filter bleibt somit trotz vermutlich relativ grosser Rechenkosten weiterhin eine hoffnungsträchtige Methode für konvektive Skalen.

Abstract

Convective phenomena in the atmosphere, such as convective storms, are characterized by very fast, intermittent and seemingly stochastic processes. They are thus difficult to predict with Numerical Weather Prediction (NWP) models, and difficult to estimate with data assimilation methods that combine prediction and observations. In this thesis, nonlinear data assimilation methods are tested on two idealized convective scale cloud models, developed in [58] and [59]. The aim of this work was to apply the *particle filter*, a method specifically designed for nonlinear models, to the two toy models that resemble some properties of convection. Potential problems and characteristic features of particle filter methodology were analyzed, having in mind possible tests on far more elaborate NWP models.

The first model, the stochastic cloud model, is a one-dimensional birth-death model initialized by a Poisson distribution, where clouds randomly appear or die independently from each other on a set of grid-points. This purely stochastic model is physically unrealistic, but since it is highly nonlinear and non-Gaussian, it contains minimal requirements for representing main features of convection. The derivation of the transition probability density function (PDF) of the stochastic cloud model made it possible to understand better the weighting mechanism involved in the particle filter. This mechanism, which associates a weight to particles (state vectors) according to their likelihood with respect to observations and to their evolution in time, is followed by *resampling*, where particles with high probability are replicated, and others eliminated. The ratio between magnitudes of the observation probability distribution and the transition probability is shown to determine the selection process of particles at each time step, where data and prediction are combined. Further, a strong sensitivity of the filter to the observation density and to the speed of the cloud variability (given by the cloud life time) is demonstrated. Thus, the particle filter can outperform some simpler methods for certain observation densities, whereas it does not bring any improvement when using others. Similarly, it leads to good results for stationary cloud fields while having difficulties to follow fast varying cloud fields, because any change in the model state variable is potentially penalized. The main difficulty for the filter is the fact that this model is discrete, while the filter was designed for data assimilation of continuous fields.

However, by representing an extreme testbed for the particle filter, the stochastic cloud model shows the importance of the observation and model error densities for the selection of particles, and it stresses the influence of the chosen model parameters on the filter's performance.

The second model considered was the modified shallow water model. It is based on the shallow water equations, to which is added a small stochastic noise in order to trigger convection, and an equation for the evolution of rain. It contains spatial correlations and is represented by three dynamical variables - wind speed, water height and rain concentration - which are linked together. A reduction of the observation coverage and of the number of updated variables leads to a relative error reduction of the particle filter compared to an ensemble of particles that are only continuously pulled (nudged) to observations, for a certain range of nudging parameters. But not surprisingly, reducing data coverage increases the absolute error of the filter. We found that the standard deviation of the error density exponents is a quantity that is responsible for the relative success of the filter with respect to nudging-only. In the case where only one variable is assimilated, we formulated a criterion that determines whether the particle filter outperforms the nudged ensemble. A theoretical estimate is derived for this criterion. The theoretical values of this estimate, which depends on the parameters involved in the assimilation set up (nudging intensity, model and observation error covariances, grid size, ensemble size,...), are roughly in accordance with the numerical results. In addition, comparing two different nudging matrices that regulate the magnitude of relaxation of the state vectors towards the observations, showed that a diagonally based nudging matrix leads to smaller errors, in the case of assimilating three variables, than a nudging matrix based on stochastic errors added in each integration time step.

We conclude that the efficient particle filter could bring an improvement with respect to conventional data assimilation methods, when it comes to the convective scale. Its success, however, appears to depend strongly on the parameters of the test setting.

Chapter 1

Introduction

Although Numerical Weather Prediction (NWP) has been progressing fast over the last decades, the formation and propagation of convective systems, responsible for severe weather such as heavy precipitation, gust fronts or strong winds can still only be poorly predicted [5]. Weather prediction for the convective scale is based on numerical models with horizontal resolution currently ranging from 1.33 km (NCEP) and 1.5km (MetOffice), over 2.5km (Météo France) to 2.8km (DWD), allowing convective storms to form and evolve explicitly, without parameterized convection. Beneath the difficulties of coupling convective scales to the weather dynamics of larger synoptic scales and the uncertainties related to the parametrization, boundary and initial conditions, a fundamental problem of NWP consists of simulating the life cycle of convective cells, which appears to be stochastic in nature [12],[37],[5].

Operational NWP relies on *data assimilation* [13],[28], which incorporate observations that are often sparse in space and time into NWP models. Since both model and observations contain stochastic errors, it is only possible to calculate, at each time step of new arriving observations, an *estimate* of an atmospheric state vector (called *analysis*), as well as the uncertainty that it contains. The analysis serves as initial condition for the consecutive *forecast*, and both analysis and forecast are then sequentially computed forward in time. Data assimilation is challenging due to the large number of unknowns (for a typical NWP model it is of the order 10^9), the nonlinear dynamics of the atmosphere, the errors in both the observations and the NWP model that are not necessarily Gaussian or uncorrelated with each other. This is why current data assimilation methods used in operational NWP encounter difficulties. Unlike for synoptic scales where linear balances dominate the system [2], when it comes to convective scale data assimilation, current models loose track of the fast varying and apparently unpredictible phenomena.

The focus of this work lies on convective scale data assimilation. Convective phenomena (cf. Figures 1.2, 6.1) are characterized by non-Gaussian probability density functions of the model and observation errors and strong non-linearities of the model (see e.g. [48]). They display an inherent random behaviour which is difficult to track by forecast models.

An effect can be seen for example on precipitation radars (cf. Figure 1.1), where initially randomly distributed, widespread isolated convective cells suddenly merge, from one observation time to another, to a large conglomerate [24]. Since radars give not enough information before convection is established and since observations are sparse at these scales, a significant error growth between observation times, as well as strong intermittency, can be observed [58], [60].

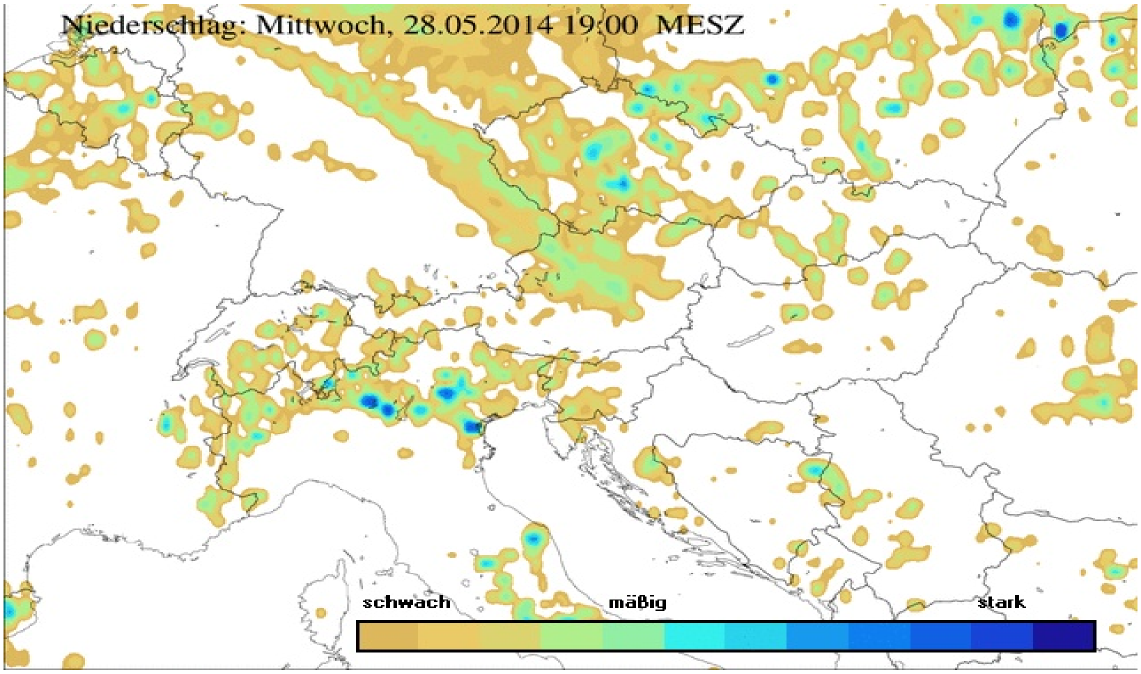


Figure 1.1: Rain reflectivity radar, taken from ZAMG (Zentralanstalt für Meteorologie und Geodynamik), the Austrian weather forecast service.

Most methods of data assimilation were developed for the synoptic scale. They are based on linear estimation theory [44],[51] and assume, for simplicity, Gaussian additive errors for the model, observations and background error statistics. The 3DVAR [35] (in operation at DWD for their global model) method computes the estimate of a state by sequentially maximizing its (Gaussian) likelihood with respect to the observations. This is done by minimizing the negative exponent of the likelihood using all observations of a time window simultaneously. The uncertainty of the initial condition, represented by the background error covariance matrix, is constant in 3DVAR and has the advantage of smoothening boundaries between regions with different observations [26]. In 4DVAR [53] (operational at Météo France and at ECMWF, the European Centre for Medium-Range Weather Forecasts), the uncertainty is time dependent, which leads to more consistent analyses [28]. But it renders 4DVAR numerically expensive [25], because computing the analysis requires integrating both tangent linear model and the adjoint of the NWP model. In case of strong non-linearity and non-Gaussianity of the errors, the methods loose their efficiency [28].

The ensemble-based methods have been investigated in the last years in order to cope better with the issues of computational cost and nonlinearity of models. One of them is the Ensemble Kalman filter (EnKF) [23], originating from the Kalman filter (KF) [38], itself based on stochastic filtering used in engineering. The KF recursively computes the state estimate and the error uncertainty, driving the focus on the stochastic nature of the model error. Still assuming Gaussian error statistics and linearized models, it allows an explicit computation of the covariance errors at each time step. The Ensemble Kalman Filter (EnKF) differs from the KF in that it propagates an *ensemble* of realizations of the *non-linear* model in time and computes an error covariance based on deviations from the ensemble's *mean* [18]. The EnKF therefore quantifies uncertainty only in the space spanned by the ensemble [25], which makes it inaccurate due to the small dimension [25]. In addition, spurious correlations caused by the limited sample size propagate over distant locations in the uncertainties [22],[28]. Refined versions of the EnKF such as the Local Ensemble Transform Kalman filter (LETKF), which assimilates observations from a local region surrounding the analysis [25], are currently being explored for convective scale applications [23], [40]. The German weather service (DWD) for example has developed a 'Km-Scale Ensemble-based Data Assimilation' (KENDA) system for their mesoscale model (COSMO), which is seen as promising especially for convective scale applications. The main advantages of the EnKF are that it provides an explicit estimate of the forecast and analysis error covariances, and that it is simpler to implement than 4D-Var [28], since it does not need an adjoint model.

Figure 1.2: Picture of a cumulonimbus in the Auvergne in central France, by Florent Courty (taken from http://www.ouest-orages.org).

However, although these methods are promising especially in combination with each other, they still rely on Gaussian error assumptions and on high dimensional forecast error covariances. This makes them unsatisfactory in severe weather situations where non-Gaussianity and nonlinearity have to be taken into account [52], and where forecasts have to be computed fast.

In a context of ongoing research more and more directed towards nonlinear data assimilation methodologies, this work aims at testing new methods specially designed to cope with seemlingly stochastic, fast varying (weather) situations in the frame of convective scale NWP. One of them is the particle filter [36], which in recent years has experienced a rapid development [32], [33], [2]. Particle filters are a sequential Monte-Carlo method which samples an ensemble of model forecasts to approximate the posterior by a sum of states weighted by observations, and evolve it in time. Sampling is a necessary solution to the problem of integrating over huge dimensions to get expectation values of atmospheric states. Unlike the methods mentioned above, particle filters aim at calculating the *full pdf* of the posterior and cannot restrict themselves to the first two moments anymore, as former *close-to-linear* model assumptions made it possible. Since nonlinear dynamics do not preserve Gaussian statistics, the main problem in nonlinear filtering is the absence of knowledge of the error statistics, and choosing a representative sample for the approximation of the posterior remains an important challenge.

A well-known problem of the particle filter was rapidly recognized in the so-called *degeneracy* of the filter [14],[6], by which a rapid depletion of the ensemble during model integration is meant, due to the fact that most particles (state vectors) become highly unprobable while only one particle gathers all the probability 'weight'. In [49] it was shown that it was impossible to avoid this phenomenon for large dimensional observation $\mathbf{d}^t$ densities, unless the ensemble size k increases exponentially with the variance of the observation log-likelihood. In other words, collapse of the filter takes place if

$$k \ll e^{\frac{\tau^2}{2}}, \tag{1.1}$$

where

$$\tau^2 := var\left(-\sum_{j=1}^{m} \log p(d_j^t|\mathbf{x}_{i,j}^t)\right) \tag{1.2}$$

is a measure for the variability of the observation likelihood for each ensemble member $\mathbf{x}_i$, and m is the observations' dimension.

Different methods have been investigated in the last years to mask the problem of degeneracy [3], [7], [14], among which the *equivalent weight particle filter* [32], [33], [2]. The equivalent weight particle filter is based on an *improved sampling* of the particles at each time step, according to a new *importance density*, q. This importance density increases the particles' likelihood by 'pushing' them closer to observations via the modified model integration equation [33]. At the last time step of an assimilation cycle, it is judiciously chosen such that the particles become equiprobable, before importance resampling replicates highly probable particles and discards the others [4], [32]. In this way, more than one particle have a chance to be re-selected for a consecutive assimilation cycle, and the ensemble of particles stays multiple and thus, minimally representative for a pdf.

This thesis presents the results of applying versions of the efficient particle filter [33], [1] on two one-dimensional toy models [58], [59] that simulate essential features of cumulus convection. The first model, the stochastic cloud model [58], is a *discrete* birth death process of clouds and *purely stochastic*. The second, modified shallow water model [59], is physically more realistic since it is described by shallow water equations which are stochastically perturbed. The latter includes simulated up- and downdraft forces that vehiculate gravity waves, which are present in convective cloud life cycles. Both models, although they do not aim to be realistic [58], [59], [57], were meant to serve as minimal testing tools for convective data assimilation methods.

The aim of this work was to test, for the first time, data assimilation methods *specially designed* for models containing *strong nonlinearities*, *stochastic* and *non-Gaussian* features, on this hierarchy of convective scale toy-models developed in [58], [59]. The efficient particle filter was, in the current art of knowledge, a natural candidate for this undertaking. The intention was neither to compare to, nor to reproduce results of [58], [59], in which conventional data assimilation methods were tried out, but to continue along their lines and give - as far as possible - complementary information about problems that may arise in a future application of particle filters to much more complex test models such as NWP models.

The stochastic cloud model [58] is a challenge to the particle filter since it is purely stochastic and contains no physical dynamics, and no spatial correlations. Although it is very unrealistic from a physical point of view, Chapter 4 reveals that this model contains interesting information with respect to the *functioning* of the particle filter. The particle filter is very sensitive to the observation density *as compared* to the model error density, and to the *speed of the variability* of the stochastic process. To understand the mechanism of these sensitivities was only possible by deriving the *exact mathematical probability density* function of the model, which is usually impossible in NWP. It also turns out, however that, because the model is discrete, the filter in [33] based on a *gradual relaxation* of the state towards observations is indirectly in contradiction with integer-valued model states.

In contrast to the simple stochastic cloud model, the shallow water model [59] tested in Chapter 5 is far more realistic since it contains, in addition to spatial correlations of the perturbations, physical evolution dynamics given by partial differential equations for three interconnected variables, wind speed, water height and rain concentration. Here, the gradual relaxation in the efficient particle filter can be fully exploited and gives satisfactory results for certain combinations of parameter settings. The hypothesis is confirmed that *only decreasing assimilation information* in terms of observations or assimilated variables is likely to display a significant benefit of the particle filter over simple 'nudging', the simplest data assimilation algorithm based on relaxation to observations. Another test reveals that the particle filter is very sensitive to the relaxation matrix used. In a further step, it is experimentally shown that there is a quantity that gives an *a priori* information about the potential success of the filter over pure relaxation. This quantity is the *standard deviation* of the (exponent of the) observation error density, and it will be theoretically derived, together with the other density exponents that compete with the observation density, and roughly estimated in the case of assimilating one dynamical, variable in Chapter 5. This emphasizes again, as in the previous model, the *relative* importance of the observation density compared to the model error density. In this context, it is shown that here as well as in the former toy model, the parameters involved in the error assumptions and in the setting of the filter have a strong impact on the resulting performance of the filter. Finally, the effect, on the standard deviations of the different densities involved in the weight exponents, of equalizing the weights before the resampling step, is briefly analyzed. This gives an indication why equalizing weights improves the selection of well placed particles with respect to observations.

In Chapter 2, a short overview of the theory of stochastic filtering is given. Chapter 3 presents the two toy models [58] and [59] for convection and outlines the methodology of the particle filter based on [33], that was applied to them. Chapters 4 and 5 show the results of the application of the particle filter on the stochastic cloud model and on the modified shallow water model, respectively. The Conclusion (Chapter 6) recapitulates the results and opens an outlook to further problems and issues on future applications of the particle filter.

Chapter 2

Basic Principles of stochastic filtering

2.1 Problem setting

The filtering problem consists of computing the conditional probability distribution of an atmospheric state vector at present time given a set of observations from time 0 up to the current time [4]. Filtering is a special case of a hidden Markov model. Hidden Markov models were developped in the frame of signal theory to describe stochastic systems that are observed through noisy measurements, where the goal is to reconstruct the original signal $(\mathbf{x}_t)_{t>0}$ from noisy observations $(\mathbf{d}_t)_{t>0}$.

For stating the general problem, let $(\mathbf{x}_t)_{t\geq 0}, t \in \mathbb{N}$, be a discrete time sequence of signal or 'state' values and $(\mathbf{d}_t)_{t\geq 0}$ a sequence of measured observations such that

$$\begin{cases} \mathbf{x}_t = f(\mathbf{x}_{t-1}, \alpha_t), \quad \mathbf{x}_0 := \mathbf{x}_0 \\ \mathbf{d}_t = h(\mathbf{x}_t, \beta_t), \quad \mathbf{d}_0 = h(\mathbf{x}_0, \beta_0) \end{cases}, \tag{2.1}$$

where α_t, β_t are independent, identically distributed (iid) sequences of random variables that represent the stochastic uncertainty in the model and the measurements, respectively. Usually, α_t and β_t are Brownian motions, independent of $\mathbf{x}_t$. The measurable function f and h stand for the model and the *observation process* respectively. They are deterministic and either linear, or nonlinear.

The goal is to determine the conditional distribution of the state vector $\mathbf{x}_t$ at time t given the information accumulated from observing $(\mathbf{d}_t)_{t\in\mathbb{N}}$ in the interval $[0, t]$, namely

$$\mathbf{x}_t^a := E[\mathbf{x}_t | \{\mathbf{d}_t, ..., \mathbf{d}_0\}]. \tag{2.2}$$

2.2 The limits of linear filtering

One of the main filters developped for a situation where the dynamical system is sufficiently well described by a linear model (f, h), perturbed by an independent, Gaussian noise, was the *Kalman filter* ([38],[39]).

2.2.1 Kalman filter

In the special case where f and h are linear and where the uncertainty is an additive independent, identically distributed, Gaussian white noise, the model and observation equation in (2.1) can be formulated as

$$\begin{cases} \mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \boldsymbol{\alpha}_t \\ \mathbf{d}_t = \mathbf{H}_t\mathbf{x}_t + \boldsymbol{\beta}_t \end{cases}, \tag{2.3}$$

where $f =: \mathbf{F} \in \mathbb{R}^{n\times n}$, $h =: \mathbf{H} \in \mathbb{R}^{m\times n}$ $(m \leq n)$ are matrices, $\mathbf{x}_t \in \mathbb{R}^n$ and $\mathbf{d}_t \in \mathbb{R}^m$, and where $\boldsymbol{\alpha}_t \in \mathbb{R}^n$ and $\boldsymbol{\beta}_t \in \mathbb{R}^m$ are mutually independent, identically distributed Gaussian noise processes with mean 0 and covariance matrices $\mathbf{Q}_t \in \mathbb{R}^{n\times n}$, $\mathbf{R}_t \in \mathbb{R}^{m\times m}$ respectively,

$$\boldsymbol{\alpha}_t \sim \mathcal{N}(0, \mathbf{Q}_t), \quad \boldsymbol{\beta}_t \sim \mathcal{N}(0, \mathbf{R}_t). \tag{2.4}$$

Under these conditions, the Kalman Filter produces an optimal estimate. The optimality of the Kalman filter in time relies on the assumption that, at any given time t, the estimate of $\mathbf{x_t}$ has Gaussian statistics. This allows the Kalman filter to recursively propagate in time estimates of only the mean and the covariance, and not of higher moments.

Kalman filter is divided into two steps, a *forecast step* and an *analysis step.* In the first step, the forecast step, the mean model state is evolved according to the linear model equation (2.3). From this equation, the mean and covariance of the state space at time t, the so-called *forecast* and *forecast error covariance*, both conditioned on the former observations (up to $t-1$), can be determined by

$$\mathbf{x}_t^f := E[\mathbf{x}_t|\mathbf{d}_{t-1}] = \mathbf{F}_t E[\mathbf{x}_{t-1}|\mathbf{d}_{t-1}] \tag{2.5}$$

$$\mathbf{P}_t^f := \mathbf{Q}_{t-1} + \mathbf{F}_{t-1}\mathbf{P}_{t-1}^a\mathbf{F}_{t-1}^T. \tag{2.6}$$

Here,

$$E[\mathbf{x}_{t-1}|\mathbf{d}_{t-1}] =: \mathbf{x}_{t-1}^a \tag{2.7}$$

is the *analysis* at time t-1, and $\mathbf{P}_{t-1}^a$ the *analysis error covariance.* At the initial beginning of the time recurrence, the *first guess* $\mathbf{x_0^f}$ is also supposed Gaussian. If $\mathbf{x}_{t-1}$ is Gaussian, then, since F is linear, $\mathbf{x}_t$ is Gaussian with mean $\mathbf{x_t^f}$ and covariance matrix $\mathbf{P_t^f}$.

The analysis step is obtained assuming the above quantities, $\mathbf{x}_t^f$ and $\mathbf{P}_t^f$, to be given. Observation and forecast are combined in this step to obtain the analysis and the analysis error covariance matrix. Let us start with the Bayesian theorem. We have

$$p(\mathbf{x}_t|\mathbf{d}_t) = p(\mathbf{x}_t|\mathbf{d}_t, \{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\}) \tag{2.8}$$

$$= \frac{p(\mathbf{d}_t|\mathbf{x}_t, \{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\})p(\mathbf{x}_t| \{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\})}{p(\mathbf{d}_t| \{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\})} \tag{2.9}$$

using repeatedly the definition of conditional probabilities, as e.g. in [10]. Let us replace $\{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\}$ by $\{\mathbf{d}_{t-1}\}$ for simplicity of notation in what follows, and leave out former observations in the conditioning.

The analysis step in the Kalman Filter is derived by showing that the pdf $p(\mathbf{x}_t|\mathbf{d}_t)$ in equation (2.8) is a multivariate Gaussian. To this purpose, all densities involved in expression (2.9) have to be evaluated. The observation density $p(\mathbf{d}_t|\mathbf{x}_t)$ is Gaussian,

$$\mathbf{d}_t|\mathbf{x}_t, \mathbf{d}_{t-1} \sim \mathcal{N}(\mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t), \tag{2.10}$$

due to the Gaussian assumption in (2.3) and (2.4). The density $p(\mathbf{x}_t| \{\mathbf{d}_{t-1}, ..., \mathbf{d}_0\})$ is Gaussian since from (2.5), we have

$$\mathbf{x}_t|\mathbf{d}_{t-1} \sim \mathcal{N}(\mathbf{x}_t^f, \mathbf{P}_t^f), \tag{2.11}$$

And finally, we show that the density $p(\mathbf{d}_t|\mathbf{d}_{t-1})$ is Gaussian distributed. A Gaussian distributed random variable is characterized by its mean and its covariance. From equation (2.3), we compute the mean of $\mathbf{d}_t$ given $\mathbf{d}_{t-1}$.

$$E[\mathbf{d}_t|\mathbf{d}_{t-1}] = E[\mathbf{H}_t\mathbf{x}_t + \boldsymbol{\beta}_t|\mathbf{d}_{t-1}] = \mathbf{H}_t\mathbf{x}_t^f. \tag{2.12}$$

Similarly, the covariance follows from

$$\begin{aligned}
\mathrm{Cov}[\mathbf{d}_t|\mathbf{d}_{t-1}] &= E[(\mathbf{d}_t - E[\mathbf{d}_t|\mathbf{d}_{t-1}])\,(\mathbf{d}_t - E[\mathbf{d}_t|\mathbf{d}_{t-1}])^T\,|\mathbf{d}_{t-1}] \\
&= E[(\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t^f)(\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t^f)^T|\mathbf{d}_{t-1}] \\
&= E[(\mathbf{H}_t(\mathbf{x}_t - \mathbf{x}_t^f) + \boldsymbol{\beta}_t)(\mathbf{H}_t(\mathbf{x}_t - \mathbf{x}_t^f) + \boldsymbol{\beta}_t)^T|\mathbf{d}_{t-1}] \\
&= \mathbf{H}_t E[(\mathbf{x}_t - \mathbf{x}_t^f)(\mathbf{x}_t - \mathbf{x}_t^f)^T|\mathbf{d}_{t-1}]\mathbf{H}_t^T + E[\boldsymbol{\beta}_t\boldsymbol{\beta}_t^T|\mathbf{d}_{t-1}] \\
&= \mathbf{H}_t\mathbf{P}_t^f\mathbf{H}_t^T + \mathbf{R}_t, \qquad (2.13)
\end{aligned}$$

and

$$\mathbf{d}_t|\mathbf{d}_{t-1} \sim \mathcal{N}(\mathbf{H}_t\mathbf{x}_t^f, \mathbf{H}_t\mathbf{P}_t^f\mathbf{H}_t^T + \mathbf{R}_t). \tag{2.14}$$

From (2.10), (2.11) and (2.14), it finally follows that (2.8) can be written as

$$p(\mathbf{x}_t|\mathbf{d}_t) = ce^{-\frac{1}{2}\mathbf{J}}, \tag{2.15}$$

where

$$\begin{aligned}
\mathbf{J} = (\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t)\mathbf{R}^{-1}(\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t) + (\mathbf{x}_t - \mathbf{x}_t^f)\mathbf{P}_t^f(\mathbf{x}_t - \mathbf{x}_t^f) \qquad (2.16) \\
- (\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t^f)(\mathbf{H}_t\mathbf{P}_t^f\mathbf{H}_t^T + \mathbf{R})^{-1}(\mathbf{d}_t - \mathbf{H}_t\mathbf{x}_t^f)
\end{aligned}$$

and where c is the constant

$$c = (2\pi)^{-\frac{n}{2}}(|\mathbf{R}||\mathbf{P}^f|)^{-\frac{1}{2}}|\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}|^{\frac{1}{2}}. \tag{2.17}$$

with |.| denoting the matrix determinant. **J** and c can be reformulated after introducing the expression

$$\mathbf{P^a} := ((\mathbf{P^f})^{\mathbf{-1}} + \mathbf{H^T R^{-1} H})^{\mathbf{-1}} \tag{2.18}$$

and using some matrix manipulation as in ([10]), as

$$\begin{aligned} \mathbf{J} &= ((\mathbf{x}_t - \mathbf{x}_t^f) - \mathbf{K}(\mathbf{d}_t - \mathbf{x}_t^f))^T (\mathbf{P}^a)^{-1}((\mathbf{x}_t - \mathbf{x}_t^f) - \mathbf{K}(\mathbf{d}_t - \mathbf{x}_t^f)) \\ c &= (2\pi)^{-\frac{n}{2}} |\mathbf{P}^a|^{-\frac{1}{2}}, \end{aligned} \tag{2.19}$$

where

$$\mathbf{K}_t = \mathbf{P}_t^f \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_t^f \mathbf{H}_t + \mathbf{R}_t)^{-1} \tag{2.20}$$

is the *Kalman gain matrix*. This formulation shows that the posterior $p(\mathbf{x}_t|\mathbf{d}_t)$ is Gaussian with mean and covariance

$$\mathbf{x}_t^a := E[\mathbf{x}_t|\mathbf{d}_t, ..., \mathbf{d}_1] = \mathbf{x_t^f} + \mathbf{K_t}(\mathbf{d_t} - \mathbf{H_t x_t^f}) \tag{2.21}$$

$$\mathbf{P}_t^a := \mathrm{Cov}[\mathbf{x}_t|\mathbf{d}_t] = (\mathbf{Id} - \mathbf{K}_t \mathbf{H}_t)\mathbf{P}_t^f \tag{2.22}$$

respectively. $\mathbf{x}_t^a$ and $\mathbf{P}_t^a$ constitute the new analysis and analysis error covariance, which completes the solution of the Kalman filter problem setting.

This is the optimal solution [3] of the Gaussian filter, because the analysis and the covariance matrix are such that the likelihood of the state given the observations becomes maximum, i.e. the *cost function* **J**, which is a quadratic function under the assumption of linear models and Gaussian errors, becomes minimal [26]. When these restrictive conditions of Gaussianity and linearity are not given, the filter is no longer optimal.

2.2.2 Ensemble Kalman Filter

A step towards nonlinear filtering was done with the Ensemble Kalman Filter (EnKF), where the assumptions of a Gaussian model and observation error are made but where the model and the observation operator may be nonlinear [25].

Why an ensemble ?

The EnKF applies a so-called Markov Chain Monte Carlo method to solve the Fokker Planck equation, which describes the evolution in time of the probability density function of a model state. This probability density can be represented using a large ensemble of model states [16].

Unlike in former versions of Kalman filters (like in the Extended KF, [26]), there is no need of linearization of the nonlinear trajectory or building up a computationally expensive 'tangent linear model'. But as in the linear case of Kalman filtering, the model error and the error of the observations are Gaussian, and the procedure of an assimilation in two consecutive steps of computing first the forecast and the forecast error covariance, then the analysis and the analysis covariance, is the same. Evensen ([16]) has shown that this ensemble representation is equivalent to the usual KF representation of the first two moments.

In the forecast step, it is an ensemble of, say k, members, which will be evolved from time t-1 to time t.

$$\begin{cases} \mathbf{x}_t^i = f(\mathbf{x}_{t-1}^i) + \boldsymbol{\alpha}_t^i \\ \mathbf{d}_t^i = h(\mathbf{x}_t^i) + \boldsymbol{\beta}_t^i \end{cases} \quad , \quad i \in [1, ..., k] \tag{2.23}$$

where f, h are now nonlinear functions. $\boldsymbol{\alpha}_t^i$ and $\boldsymbol{\beta}_t^i$ are - as in the linear case - mutually independent Gaussian noise processes.

The error covariance of the forecast is now not derived from the Gaussian error statistics as in equation (2.5), which was based on the deviation from an unknown truth, but now is based on the deviation from the ensemble mean ([16]). The underlying assumption of this choice is that the ensemble mean is the best estimate and that the spreading of the ensemble around the mean is interpreted as the error in the ensemble mean.

With the same notation as before, the forecast and forecast error covariance are now calculated from the ensemble as

$$\mathbf{P}_t^f = \frac{1}{k-1} \sum_{i=1}^{k} (\mathbf{x}_t^f - \bar{\mathbf{x}}_t^{f,i})(\mathbf{x}_t^{f,i} - \bar{\mathbf{x}}_t^{f,i})^T \tag{2.24}$$

$$\bar{\mathbf{x}}_t^f = \frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_t^{f,i}. \tag{2.25}$$

The analysis is computed for each member $i \in [1, k]$ as before in equation (2.21)

$$\mathbf{x}_t^{a,i} = \mathbf{x}_t^{f,i} + \mathbf{K}_t(\mathbf{d}_t + \eta_t^i - \mathbf{H}\mathbf{x}_t^{f,i}), \tag{2.26}$$

except that now the observations are perturbed with a Gaussian noise $\eta_t^i \sim \mathcal{N}(0, \mathbf{R})$. The Kalman gain matrix $\mathbf{K_t}$ equals the one in (2.20).

The same as for the forecast step holds for the analysis error covariance and the analysis [9],

$$\mathbf{P}_t^a = \frac{1}{k-1} \sum_{i=1}^{k} (\mathbf{x}_t^{a,i} - \bar{\mathbf{x}}_t^{a,i})(\mathbf{x}_t^{a,i} - \bar{\mathbf{x}}_t^{a,i})^T \tag{2.27}$$

$$\bar{\mathbf{x}}_t^a = \frac{1}{k} \sum_{i=1}^{k} \mathbf{x}_t^{a,i}. \tag{2.28}$$

The essential feature of the EnKF is that, unlike in the KF where the uncertainty of the state estimate was evolved in time via the corresponding covariance matrix, it evolves an uncertainty based on the number of ensemble members. Therefore, the rank of the covariance matrix of formerly the system's dimension is now reduced to the ensemble's size minus one, k-1.

The problem of this rank reduction of the forecast uncertainty matrix is that fast and chaotic features of the system may not be captured by the assimilation, and errors may grow. Due to this dimensional 'undercoverage' of the state covariance, the uncertainty of the state is underestimated so that the assimilation fails in tracking the the 'truth' and therefore, the observations [56].

In order to counterbalance this effect of underrepresentation of the errors, it is necessary to localize the area where the analysis is undertaken and to introduce a (multiplicative or additive) parameter in order to artificially increase the analysis covariance matrix [25]. Localization permits additionally to avoid suprious correlations between distant locations in the background covariance matrix $\mathbf{P}_t^f$. Of course, increasing the ensemble size leads to a decrease of the errors in the Monte Carlo samplings, proportional to $1/\sqrt{k}$ [16].

Remark

Up to now, the EnKF is - together with 4DVAR - the most developed and efficient method in data assimilation [17]. The main advantages of the EnKF are that it provides an estimate of the forecast and analysis error covariances, and that it is much simpler to implement than 4D-Var, since it can be parallelized. It has no tangent linear model or adjoint operator, no integration backward in time [16]. But it has to deal with the problem of assimilating observations only in a (much) lower dimensional space.

The Kalman filter uses a linear model and Gaussian error assumptions to obtain a Gaussian posterior, which is perfectly described by the first two moments of the pdf. The EnKF in contrast uses a nonlinear model and Gaussian error statistics to compute the first two moments. For strong nonlinear models, however, the first two moments do not suffice to describe the pdf of the state anymore.

In case of rapidly evolving weather regimes, these limitations become a serious handicap. Nonlinear filtering, which is not limited to the computation of the first two moments of the pdf anymore, offers solutions for highly non-Gaussian and nonlinear dynamics as they are encountered in weather prediction.

2.3 Nonlinear filtering

In nonlinear filtering, we wish to estimate the *full probability density function* (pdf) of a state vector $\mathbf{x}_t$. In order to do so, the pdf $p(\mathbf{x}_t|\mathbf{d}_t, ..., \mathbf{d}_{t-L})$ conditioned on all preceding observations up to the present one, $\mathbf{d}_{t-L}$,...,$\mathbf{d}_t$ ($L \leq t$), is computed recursively. $p(\mathbf{x}_t|\mathbf{d}_t, ..., \mathbf{d}_{t-L})$ is called 'posterior' density, while $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the 'prior' and is available at the beginning of each time-iteration of the computation. In particular, $p(\mathbf{x}_0|\mathbf{d}_0) := p(\mathbf{x}_0)$. Let us estimate the posterior and start with equation (2.8) using *Bayes Theorem.* We have

$$p(\mathbf{x}_t|\mathbf{d}_t) = \frac{p(\mathbf{d}_t|\mathbf{x}_t, \{\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L}\})p(\mathbf{x}_t|\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L})}{p(\mathbf{d}_t|\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L})}. \tag{2.29}$$

If observations $\mathbf{d}_{t-1}$ at time $t-1$ are not available, $p(\mathbf{x}_t|\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L})$ in (2.29) can be expressed in terms of the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the probability of the former state conditioned on preceding observations through the *Chapman-Kolmogorov* equation for Markov chains,

$$p(\mathbf{x}_t|\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L}) = \int_{\mathbb{R}^n} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{d}_{t-2}, ..., \mathbf{d}_{t-L})d\mathbf{x}_{t-1} \tag{2.30}$$

(for an n-dimensional state vector $\mathbf{x}_{t-1}$). Equation (2.30) implies the *Markov property,*

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L}), \tag{2.31}$$

stating that a Markov process is independent of all but the last former timesteps ('forgetting the past'). The posterior (2.29) can now be written as

$$p(\mathbf{x}_t|\mathbf{d}_t) = \frac{p(\mathbf{d}_t|\mathbf{x}_t, \{\mathbf{d}_{t-2}, ..., \mathbf{d}_{t-L}\})}{p(\mathbf{d}_t|\mathbf{d}_{t-2}, ..., \mathbf{d}_{t-L})} \int_{\mathbb{R}^n} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{d}_{t-2}, ..., \mathbf{d}_{t-L})d\mathbf{x}_{t-1},$$

where $p(\mathbf{d}_t|\mathbf{d}_{t-1}, ..., \mathbf{d}_{t-L})$ is a constant. The Chapman- Kolmogorov equation can be reiterated for as many timesteps as no previous observations are available until the prior of the previous assimilation time step appears in the integral expression (2.30). If new observations are available every L timesteps, then substituting the resulting L-multiple integral into

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) = \int_{\mathbb{R}^n} f(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{d}_t)d\mathbf{x}_t \tag{2.32}$$

leads to the (L+1)-multiple integral

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) = \frac{1}{p(\mathbf{d}_t)} \underbrace{\int_{\mathbb{R}^n} ... \int_{\mathbb{R}^n}}_{\text{L+1}} f(\mathbf{x}_t)p(\mathbf{d}_t|\mathbf{x}_t)\left(p(\mathbf{x}_t|\mathbf{x}_{t-1})...p(\mathbf{x}_1|\mathbf{x}_{t-L})\right)p(\mathbf{x}_{t-L})d\mathbf{x}_t...d\mathbf{x}_{t-L},$$

and hence to

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) = \frac{1}{p(\mathbf{d}_t)} \underbrace{\int_{\mathbb{R}^n} ... \int_{\mathbb{R}^n}}_{\text{L+1}} f(\mathbf{x}_t)p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-L})p(\mathbf{x}_{t-L})d\mathbf{x}_t...d\mathbf{x}_{t-L}, \tag{2.33}$$

where

$$p(\mathbf{x}_t|\mathbf{x}_{t-L}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})...p(\mathbf{x}_{t-L+1}|\mathbf{x}_{t-L}). \tag{2.34}$$

The (normalizing) constant in (2.33) is computed according to the Theorem of absolute probabilities and is given by

$$p(\mathbf{d}_t) = \int_{\mathbb{R}^n} p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t)d\mathbf{x}_t. \tag{2.35}$$

The posterior $p(\mathbf{x}_t|\mathbf{d}_t)$ in (2.32) is the complete solution to the state estimation problem (since it contains all statistical information), and first, second and higher moments can be calculated if needed. The task of sequential filtering consists then of recursively propagating the posterior density function in time and evaluating it each time that new observations become available.

The main difficulty lies in the integration over a usually high dimensional space, which cannot be obtained analytically. There is a class of particle filters, though, that can be used to approximate the posterior distribution numerically.

2.3.1 Sequential Importance Sampling (SIS)

One class of particle filters approximating the posterior is the *Sequential Importance Sampling (SIS)*, a Monte Carlo method for recursively evolving the posterior density in time. It consists of drawing a reduced number of particles at each timestep t. The initial prior $p(\mathbf{x}_{t-L})$ is then approximated by a finite discrete sum of say k particles, according to the empirical distribution

$$p(\mathbf{x}_{t-L}) = \frac{1}{k}\sum_{i=1}^{k}\delta(\mathbf{x}_{t-L} - \mathbf{x}^i_{t-L}). \tag{2.36}$$

After substituting this prior into the integral (2.33), we have

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) = \frac{1}{p(\mathbf{d}_t)} \underbrace{\int_{\mathbb{R}^n}...\int_{\mathbb{R}^n}}_{\text{L+1}} f(\mathbf{x}_t)p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-L})p(\mathbf{x}_{t-L})d\mathbf{x}_t...d\mathbf{x}_{t-L} \tag{2.37}$$

$$\approx \frac{1}{p(\mathbf{d}_t)}\sum_{i=1}^{k} \underbrace{\int_{\mathbb{R}^n}...\int_{\mathbb{R}^n}}_{\text{L}} f(\mathbf{x}_t)p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}^i_{t-L})d\mathbf{x}_t...d\mathbf{x}_{t-L+1}. \tag{2.38}$$

For each later timestep $s \in [t-L+1, t]$, the k particles are drawn from, and evolved according to the transition density

$$p(\mathbf{x}_s|\mathbf{x}_{s-1}) \sim \sum_{i=1}^{k} p(\mathbf{x}_s|\mathbf{x}^i_{s-1})\delta(\mathbf{x}_s - \mathbf{x}^i_s), \tag{2.39}$$

whereby the L remaining integrals are cancelled again by virtue of the properties of delta functions,

$$f(\mathbf{x}_0) = \int_{\mathbb{R}^n} f(\mathbf{x})\delta(\mathbf{x} - \mathbf{x}_0)d\mathbf{x}. \tag{2.40}$$

Hence, the conditional mean of $f(\mathbf{x})$ after evolution of the particles during L time steps can be approximated by a discrete sum of weighted (functions of) state vectors

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) \approx \sum_{i=1}^{k} w_t^i f(\mathbf{x}_t^i), \tag{2.41}$$

where

$$w_t^i = \frac{p(\mathbf{d}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-L}^i)}{p(\mathbf{d}_t)} \tag{2.42}$$

are the *weights* of the particles. The weights indicate how probable a particular state $\mathbf{x}_t^i$ is at the time t of evaluation of the posterior, given the observations and the previous states up to this timestep. They are normalized before a new assimilation cycle with new observations starts.

In the limit of an infinite number of particles, the sum approximation of the posterior for $L = 0$ (i.e. for observations being available at every time step) approaches the true posterior density ([3],[4]),

$$\sum_{i=1}^{k} w_i \delta(\mathbf{x}_t - \mathbf{x}_t^i) \xrightarrow[\text{k} \to \infty]{} p(\mathbf{x}_t|\mathbf{d}_t, ..., \mathbf{d}_0). \tag{2.43}$$

At each iteration step $t - 1 \to t$ of the assimilation, the particles change their respective position according to the stochastic evolution equation and end up at some new position with a probability given by the transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Their motion is entirely determined by the model equation.

2.3.2 Importance density

Sometimes, it happens that the prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is known and easy to evaluate up to a (normalizing) constant, but difficult to generate. It is well known in numerical science [54] that there is only a restricted group of probability density functions that can easily be generated numerically. It also can happen that the normalization constant is difficult to compute. In other cases, the prior and its normalizing constant are easy to compute, but the prior is so far from the observation density that the posterior,

$$p(\mathbf{x}_t|\mathbf{d}_t) = \frac{p(\mathbf{x}_t|\mathbf{d}_t)p(\mathbf{x}_t)}{p(\mathbf{d}_t)}, \tag{2.44}$$

is almost everywhere close to zero. In all these cases [1], it is necessary or preferable to generate state vectors from a new probability density function q, called *importance density* or *proposal density* ([3]).

In order to avoid changing the value of the posterior and hence the expectation value (2.33) by drawing from a new density, q is multiplied with both the numerator and the denominator in

$$p(\mathbf{x}_t|\mathbf{d}_t) = \frac{p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t)q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{d}_t)}{p(\mathbf{d}_t)q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{d}_t)}. \tag{2.45}$$

Another, mathematically more elegant way of introducing the proposal density can be found in [3]. Then, the integral (2.37) becomes

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) \approx \frac{1}{p(\mathbf{d}_t)} \sum_{i=1}^{k} \underbrace{\int_{\mathbb{R}^n} \dots \int_{\mathbb{R}^n}}_{\text{L}} f(\mathbf{x}_t) \frac{p(\mathbf{d}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-L}^i)}{q(\mathbf{x}_t|\mathbf{x}_{t-L}^i, \mathbf{d}_t)} q(\mathbf{x}_t|\mathbf{x}_{t-L}^i, \mathbf{d}_t) d(\mathbf{x}_t, ..., \mathbf{x}_{t-L+1}), \tag{2.46}$$

where q can be decomposed like p into

$$q(\mathbf{x}_t|\mathbf{x}_{t-L}^i, \mathbf{d}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{d}_t)...q(\mathbf{x}_{t-L+1}|\mathbf{x}_{t-L}^i, \mathbf{d}_t). \tag{2.47}$$

Sampling from q instead of p for all time steps but the prior's, i.e. for $t \in [t-L+1, t]$, leads to

$$E(f(\mathbf{x}_t)|\mathbf{d}_t) \approx \frac{1}{p(\mathbf{d}_t)} \sum_{i=1}^{k} f(\mathbf{x}_t^i) \frac{p(\mathbf{d}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-L}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-L}^i, \mathbf{d}_t)}, \tag{2.48}$$

where

$$w_t^i := \frac{p(\mathbf{d}_t|\mathbf{x}_t^i)p(\mathbf{x}_t^i|\mathbf{x}_{t-L}^i)}{p(\mathbf{d}_t)q(\mathbf{x}_t^i|\mathbf{x}_{t-L}^i, \mathbf{d}_t)} \tag{2.49}$$

are the new weights. They *correct* the erroneous modification of the model trajectory that is made by sampling from q.

Sampling from the importance density permits to avoid generating particles from p, for all reasons mentioned above. At the very first time step of an assimilation experiment, the particles are sampled from the first prior p (for example according to (2.36)). At all later time steps, they are generated from q. The density q is arbitrary and chosen such that it is easy to sample from it, but also in order to reduce degeneracy ([3]) (see below).

[1] personal communication of Dr. S. Winitzki

2.3.3 Problem of degeneracy

After some iterations, it is observed that only one particle has a non-zero weight, meaning that only one particle's position has a non zero probability to occur compared to the other particles. Since the weights are normalized at each timestep, it is only the relative likelihood among the particles with respect to the observation that impacts the weights, not the 'absolute' value of the (unnormalized) weights.

Snyder et al.[49] showed that it is impossible to avoid this phenomenon called *degeneracy* for large dimensional observations $\mathbf{d}_t$ and using the prior as an importance density, first for iid (identically independently distributed) observation error coordinates, then more generally for Gaussian priors and model errors. They derived a condition for particle filters with observations being available at every time step under which collapse of the weights at a given time step is inevitable. The main idea of the proof was to use the *Law of Large Numbers* to show that the maximal weight tends to 1 for increasing observation dimensions m and large ensemble sizes k,

$$w^i_{t,max} \sim p(\mathbf{d}_t|\mathbf{x}^i_t) \to 1 \tag{2.50}$$

for a particle i, which is equivalent ([49]) to

$$k \ll e^{\frac{\tau^2}{2}}, \tag{2.51}$$

where

$$\tau^2 := var\left(-\sum_{j=1}^{m} \log p(\mathbf{d}_j|\mathbf{x}_i)\right) \tag{2.52}$$

is a measure for the variability of the observation likelihood. Collapse of the filter takes place if (2.52) holds. In other words, k must increase exponentially with τ^2 to avoid degeneracy.

These results of filter collapse hold for the standard importance [49] density, where the prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is chosen to serve as importance density q. They were shown later to be also valid for the *optimal* importance density ([50]), where the importance density is the posterior. The optimal importance density was introduced by Doucet et al ([14]), who showed that for the case of available observations at every time step, it minimizes the variance of the weights. As pointed out by the authors ([49], p.4638), no resampling technique was considered so far, nor non-Gaussian priors or observation errors, nonlinear models, or other importance densities than the prior and the posterior, as well as the effects of cycling the filter over time. But various experiments seem to confirm that without any other 'trick' like weight equalization procedures ([32], see next sections), degeneracy is hardly to be controlled and unlikely to vanish in the current state of art.

Nevertheless, different methods are currently being investigated to mask this problem [3]. Suggestions to avoid degeneracy are based on a 'better' choice [2] of the importance density q ([3],[7]). Another hope lies in replacing the sampling step at each assimilation time step by *Resampling*.

2.3.4 More efficient particle filters

Resampling methods

The particle filter in its simplest form is a system of particles that is evolved in time according to the dynamics of the particles, and where the system is corrected at each timestep when new observations are available. This 'correction' takes place either by carrying forward in time the same particles as initially sampled (SIS), only evaluating the new likelihood of the particles with respect to the observations, or by 'choosing' each time new particles according to some criteria specified below. This procedure of sampling with correction is called *Re-sampling*.

Sequential Importance Resampling (SIR)

The Importance Resampling method was introduced by Gordon et al [20]. At the arrival time of new observations, each member of the ensemble is statistically replicated by a number of identical offspring proportional to its weight. This means that particles with too low weight among the ensemble are replaced by particles with high weight. Sequential Importance Resampling consists of the following steps [4].

Initialization step

At time $t = 0$, sample k particles $\mathbf{x}_0^i$ from

$$p(\mathbf{x}_0) := p(\mathbf{x}_0|\mathbf{d}_0) = \sum_{i=1}^{k} \delta(\mathbf{x} - \mathbf{x}_0), \quad i \in [1, k]. \tag{2.54}$$

The prior distribution is given by an arbitrary choice of k state vectors at time $t = 0$. This choice of samples should ideally be 'made well', i.e. it should be representative of the prior distribution, which is the main difficulty at this stage.

Iteration or correction step, $[t-1, t]$

For $t > 0$, evolve these particles according to their law, $p(\mathbf{x_i^t}|\mathbf{x_i^{t-1}})$. At time t, compute their weights according to (2.42). Replace each particle $\mathbf{x}_t^i$ by a number ξ_t^i of offspring such

[2] Until now, a 'good' choice meant that the likelihood of a state was increased by making q dependent on the present observations $\mathbf{d}_t$, via

$$q_t = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{d}_t). \tag{2.53}$$

that

$$\sum_{i=1}^{k} \xi_t^i = n. \tag{2.55}$$

a. Multinomial offspring distribution

The SIR filter assumes by definition that this number, ξ_t^i, of replicated 'survivors' for each particle has a multinomial distribution and thus has a probability density function for each $i \in [1, k]$ given by

$$f(\xi_t^i) = \frac{k!}{\prod_{i=k}^{k} k^i!} \prod_{i=1}^{k} (w_t^i)^{k^i}. \tag{2.56}$$

The multinomial distribution describes an empirical situation where one samples with replacement k times from a population of k members (here particles) according to the probability distribution given by the corresponding weights w_t^i. ξ_t^i is the number of times that the particle with resampled position $\tilde{x}_t^i$ is chosen.

The particle filter starts from an empirical measure $p(x_0)$ associated with k random particles of mass $1/n$ drawn from the former [4],

$$p(\mathbf{x}_0) = \frac{1}{n} \sum_{i=1}^{k} \delta_{\mathbf{x}_0^i}. \tag{2.57}$$

At a later timestep t, after resampling, the empirical distribution associated with the ensemble of resampled particles $\tilde{x}_t^i$ of mass $1/n$ is

$$p(\mathbf{x}_t) = \frac{1}{n} \sum_{i=1}^{k} \delta_{\tilde{\mathbf{x}}_0^i}. \tag{2.58}$$

If ξ_t^i is multinomial, then its expected value conditioned on former positions of particles up to $t-1$ is proportional to the corresponding weight

$$E[\xi_t^i | \mathbf{x}_{t-1}^i] \sim k w_t^i. \tag{2.59}$$

Furthermore, under the condition of boundedness of the covariance of ξ_t^i [3] for the offspring number ξ_t^i, $p(\mathbf{x}_t)$ can be written as

$$p(\mathbf{x}_t) = \frac{1}{n} \sum_{i=1}^{k} \xi_t^i \delta_{\tilde{\mathbf{x}}_t^i}, \tag{2.60}$$

[3] [4], p. 274

and

$$E[p(\mathbf{x}_t)|\mathbf{x}_{t-1}, ..., \mathbf{x}_0] = \frac{1}{n}\sum_{i=1}^{k} w_t^i \delta_{\tilde{\mathbf{x}}_t^i}. \tag{2.61}$$

The iteration uses $p(\mathbf{x}_{t-1})$ to obtain $p(\mathbf{x}_t)$, but no approximated priors at any former time step than $t-1$. In the SIR method, the evolved particles arriving at the assimilation timestep t are multiplied or discarded according to the magnitude of the weights. This *resampling* or *selection* step discards particles in unlikely positions and multiplies the more likely ones.

The SIR filter is quick and easy to implement. But the problem of degeneracy is not avoided. Besides the still remaining problem of degeneracy and possibly uniform replication, thus leading to an empoverishment, of the ensemble [7], this filter is also suboptimal because of the multinomial distribution of the offspring [4]. The resampling step replaces the weights by the random mass ξ_t^i/k. But since the offspring number ξ_t^i can take any number between 0 and k, *even when the position of a particle is very likely* (w_t^i is high), *this particle may have very few offspring or even none at all* [4].

b. Other possible offspring distributions

Bain ([4]) suggested another offspring distribution in which this problem of possible neglection of particles with high probability can be avoided, the *residual sampling*. Residual sampling is also suboptimal since here, particles with low weights may be overrepresented. Another not yet tested and maybe promising resampling method which avoids both these problems of underrepresenting high weighted particles and overrepresenting low weighted particles is the distribution from the *tree-based branching algorithm*, which was proved to converge to the right posterior density as well [15].

Remark

Other suggestions of improved resampling were made in recent years in order to cope with the problem of lack of diversity and degeneracy, as for example the attempt to equalize the weights [32] before resampling, or to resample only if the diversity is lower than a certain threshold [42]. Still another way to avoid degeneracy would be to use Monte Carlo Markov Chain (MCMC) simulations as Metropolis-Hastings, in which many repetitions of sampling at a given timestep would produce a new set of samples being representative of the pdf. As pointed out in [7], there would be no dependency on dimensionality with regard to the 'curse of dimensionality', but the computational cost for these methods seems very unfavorable and wouldn't change much in terms of spread and diversity [4]. Currently, the search for avoiding degeneracy of weights and depletion of ensembles is still an issue of serious concern.

[4] according to a personal communication of P.J. van Leeuwen

A more efficient particle filter, which uses a well-chosen importance density to avoid degeneracy by making a certain number of particles' weights equal before resampling, has been developed in recent years [33]. It is this particle filter that will be used here to test convective scale data assimilation in what follows (cf. next section).

Chapter 3

Model hierarchy and methodology

The aim of this work is to test particle filters on a hierarchy of testbed models that represent main features of convective scale phenomena such as non-Gaussianity in the errors, high nonlinearity and a stochastic behaviour of the model. Würsch and Craig developed two toy models in [58] and [59] as testbeds for convective scale data assimilation. In this chapter, these models are introduced and the particle filter method that was applied to them is described more in detail.

The first simple model is a discrete and purely stochastic cloud model without any component of deterministic physical dynamics, and it contains neither spatial correlations nor any dynamical balances. Therefore, it is the simplest possible representation of a convection process, meant to test data assimilation methods in their most basic form.

The second model in the hierarchy is a one dimensional shallow water model that links together wind velocity, water height and precipitation. It is physically more realistic than the first model, since the evolution of the variables is defined by deterministic partial differential equations, thus implying correlations between grid points.

Both models satisfy the requirement of a highly stochastic and nonlinear behaviour. Linear methods such as LETKF[1] do not succeed very well when being tested on such models (see eg. [58]). The particle filter was specially designed to deal with nonlinear models. It appears to be a promising method, since it sequentially simulates from the *full posterior* distribution [14] and does not suffer from erroneous linearizations. Also, there is no need of computing inverse matrices that involve a high computational cost. Filter collapse, a severe drawback of this method due to the elimination of particles with low likelihood, was addressed in the last few years in [33], [1], by choosing a special importance density just before resampling that makes most particles equally likely.

Section 3.1 summarizes the two idealized models for the simulation of convective clouds

[1] Local Ensemble Transform Kalman Filter (cf. [25]).

[58] and [59]. Section 3.2 outlines the methodology of the particle filter with equivalent weights [33], which is the assimilation method that will be used and investigated throughout Chapters 4 and 5.

3.1 Model hierarchy

3.1.1 The Stochastic Cloud model

The stochastic cloud model [58] is the first in a hierarchy of models designed to mimic the main features of convection, an extreme nonlinearity of the model for the representation of intermittency, and non-Gaussianity of the errors. It is based on the stochastic convection parametrization scheme of Plant and Craig [45],[57], and it simulates convective dynamics by a birth-death process.

The process is initiated by drawing, at each of $N = 100$ grid points separately, a number a of clouds from a Poisson distribution, whose probability density function p is given by

$$p(a) = \frac{\rho^a e^{-\rho}}{a!} \tag{3.1}$$

where ρ is the mean number of clouds per grid point. ρ is given by

$$E[a] = \rho = \frac{\lambda}{\mu}, \tag{3.2}$$

where λ is the birth probability and μ the death probability of a cloud. At later timesteps and at each grid point, either one new cloud appears with probability λ, or any of a existing clouds disappears with probability μ. Further, we have

$$\mu = 1 - 2^{-hl}, \tag{3.3}$$

where hl is the cloud half life, i.e. the time by which half of the number of clouds at a grid point have died, and λ is the birth probability of a cloud. Poisson processes can be encountered in many situations in Nature, like for example for the decay of radioactive substances [27]. Craig and Würsch [58] tested the SIR-filter and LETKF in this model.

In Section 4, the exact probability density function for the model error is derived before applying an efficient particle filter [32] in its simplest form to it, without equalizing the weights before resampling.

3.1.2 The Shallow Water model

A more realistic convective cloud life cycle

The modified shallow water model [59] was designed to investigate nonlinear and non-Gaussian feautures that characterize convection. Classical models used for testing data assimilation algorithms as the Lorenz or the quasi-geostrophic model do not reflect these properties. They are more suitable for large scale balanced flows and were designed for synoptic scale dynamics [58].

The shallow water equations describe the motion of a fluid of small depth subject to a gravitational field [29]. The solutions of the shallow water equations are gravity waves, which appear as a consequence of a small perturbation in depth of a fluid layer. Such a perturbation leads to a buoyancy forcing which restores the depth of the fluid to its mean height at a given location [24]. As the excess of mass is transferred to the adjacent location, a new perturbation is created and propagates in form of a wave. Gravity waves arise from density anomalies and temperature perturbations in the atmosphere that generate updraft and vertical downdraft motions [59]. They are a potentially essential phenomenon in convection, since they can be excited by the evolution of convective clouds [43],[21],[30]. Internal gravity waves, which propagate vertically, are not considered here.

The modified shallow water model [59] presents main characteristics of a convective cumulus cloud life cycle and represents a minimal model for gravity waves. These gravity waves are made possible by spatial correlations, which were absent in the stochastic cloud model and which make the model more physically realistic. The birth of a cloud is represented by an enhanced buoyancy at places where the water level reaches a predefined height threshold. This process of increased updraft simulates the effect of latent heat release in the atmosphere, which leads to condensation after a saturation level is reached, and to the appearance of clouds. Since clouds cannot grow indefinitely, there has to be a mechanism that removes them from the atmosphere. *Removal* of clouds is mainly accomplished by precipitation, which happens when the growing water level reaches a second, maximal threshold. Positive buoyancy then turns into negative buoyancy by complex thermodynamical phenomena. In addition, the slowing down of cloud accumulation is supported by diffusion.

To this 'basic lifecycle' of appearance and destruction of convective clouds, a background noise in the wind field is added in order to trigger new convective clouds. Stochastic triggering is applied continuously over the entire domain, and at every time step. In the absence of orography there is no favourite location for the production of clouds. In between clouds, the water height decreases as an effect of gravity waves response. This suppresses clouds in the imminent neighbourhood. After rain appears, the cloud collapses, causing gravity waves that will trigger new clouds.

The shallow water model

The classical shallow water equations [29] describe the motion of a free surface of an incompressible fluid in a gravitational field, when the depth of the fluid is small compared to the characteristic dimensions of the liquid surface. In this case, the vertical component of the fluid velocity may be neglected. The fluid is characterized, at each point, by its depth h and its velocity u. The shallow water equations are given by a set of two differential equations, the continuity equation (mass conservation, (3.5)) and the motion equation of the fluid (conservation of the force acting on the fluid, (3.4)). In the one dimensional case, u(x,t) and h(x,t) are scalar functions depending only on x and time. They satisfy

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial \phi}{\partial x} = 0 \tag{3.4}$$

$$\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} = 0. \tag{3.5}$$

The gradient of the geopotential $\phi = gh$ in (3.4), where g is the gravitational constant, represents the hydrostatic balance, i.e. the balance of forces between vertical pressure gradient and gravitational force acting on the fluid, of constant density ρ.

Essential features of a cumulus convection cycle are simulated by an enhanced updraft of the fluid, followed by downdraft in form of rain [24], [59]. This cycle is described by the variables wind speed (u), fluid height (h) and amount of rain (r). The shallow water model was modified in the following way [59].

(i) Cloud formation

Cloud formation is performed by condensation and updraft motions in the atmosphere. If the fluid level h reaches a certain threshold in height, H_c, enhanced buoyancy takes place and is generated by replacing the geopotential by a lower constant value ϕ_0, according to

$$\phi = \begin{cases} \phi_0 & if \quad h > H_c \\ gh & . \end{cases} \tag{3.6}$$

Through this modification of the geopotential ϕ, the pressure gradient is pushed up even more. H_c represents the level of free convection [59].

(ii) Cloud removal

In order to 'remove' clouds so that they do not remain indefinitely in the production process, the updraft is stopped by precipitation, and by diffusion. Positive buoyancy turns into negative buoyancy by a surplus of humidity (or water drops). If the water height level reaches a threshold H_r (which is higher than the threshold H_c) and if u has positive convergence i.e. h is still rising, rain is 'produced'. The consecutive decrease of the water level (after h reached the rain threshold) by instataneous rain production is modelled by adding rain water mass to the potential, which reduces again the buoyancy. Equation (3.4) becomes

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial (\phi + \gamma r)}{\partial x} = K\frac{\partial^2 u}{\partial x^2}, \tag{3.7}$$

where K is a diffusion constant and where γ is a constant that converts rain into a potential. ϕ is given by (3.6). An additional equation [59] for the rain water was introduced in equation (3.4),

$$\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial x} = K_r\frac{\partial^2 r}{\partial x^2} - \alpha r - \begin{cases} \beta\frac{\partial u}{\partial x}, & h > H_r \quad \text{and} \quad \frac{\partial u}{\partial x} < 0 \\ 0, & otherwise, \end{cases} \tag{3.8}$$

where β is the production rate, α the removal rate of rain and where K_r is the diffusion constant of the rain. The rain production rate is proportional to the convergence rate in order to simulate the delay before the rain appears.

Modified shallow water equations

All equations contain a diffusion term, mainly introduced for numerical reasons, but also, as mentioned before, to contribute to limiting the size of clouds during their formation [59]. The modified shallow water equations finally are

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial(\phi + \gamma r)}{\partial x} = K\frac{\partial^2 u}{\partial x^2} + \beta_u, \quad \phi = \begin{cases} \phi_0 & if \quad h > H_c \\ gh & else. \end{cases} \tag{3.9}$$

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} = K\frac{\partial^2 h}{\partial x^2} \tag{3.10}$$

$$\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial x} = K_r\frac{\partial^2 r}{\partial x^2} - \alpha r - \begin{cases} \beta\frac{\partial u}{\partial x} & h > H_r, \frac{\partial u}{\partial x} < 0 \\ 0 & otherwise. \end{cases} \tag{3.11}$$

β_u is a small stochastic forcing that was only added to the wind velocity in [59], in order to trigger perturbations and hence convection. For the purpose of our data assimilation tests, it was necessary to add a perturbation also to the other variables h and r (see next section). Further details of the cumulus cycle and of the modified Shallow water model are described in [59].

Initial and boundary conditions

Our data assimilation tests are based on the numerical implementation of the modified shallow water model of [59]. The initial condtions for u, h and r are set to constant values

$$\begin{aligned} u(x,0) &= u_0 \\ h(x,0) &= h_0 \\ r(x,0) &= r_0 \end{aligned} \tag{3.12}$$

The boundary conditions are periodic. For all $t \geq 0$, and for a number of grid points of N and a displacement Δx, we have

$$\begin{aligned} u(x + N\Delta x, t) &= u(x,t) \\ h(x + N\Delta x, t) &= h(x,t) \\ r(x + N\Delta x, t) &= r(x,t) \end{aligned} \tag{3.13}$$

In this work and unlike in [59], no orography is taken into account. The minimum value of h is zero.

3.2 Efficient Particle filter with weight equalization

In the following Sections 4 and 5, we apply the efficient particle filter based on the implementation of [33] to the convective toy models presented above. We use an Importance density that involves nudging the particles towards future available observations.

For the stochastic cloud model in Section 4, we derive the exact mathematical transition density p of the model error. We further use a Gaussian or an exponential observation density for the computation of the weights. For simplicity of implementation and out of the wish to understand the filter in its basic form, we do not include equalization of weights in the stochastic cloud model. The procedure of equalizing the weights of the particles at the last time step before resampling was introduced in [33] in order to avoid filter collapse onto one ensemble member. Since we deal with a very simple model and since we have the correct probability transition density at hand, we wish to understand under which circumstances filter collapse takes place.

In contrast, for the modified shallow water model in section 5, the implementation of the filter is fully based on [33]. The error of the truth run and of the nudged model at each integration time step is chosen to be Gaussian, as well as the observation error density. Furthermore, the equalization procedure of weights at the last time step before resampling is integrated into the filter.

The present Section specifies the crucial, practical computation steps in [33] set up of the filter for the case where all error statistics are Gaussian. This is the methodology we use for the shallow water model. Up to the error statistics and the weight equalization, the stochastic cloud model is based on the same procedure. In what follows, we define the expression for the weights when Gaussian densities are used at each assimilation time step, and remind briefly the procedure of weight equalization of [33],[1].

3.2.1 Gaussian model errors

For the shallow water model, let us define a particle $\mathbf{x}_i$ ($i \in [1, k]$) as the vector

$$\mathbf{x}_i := (\mathbf{u}_i, \mathbf{h}_i, \mathbf{r}_i) \in \mathbb{R}^{3N}, \tag{3.14}$$

where N is the number of grid points. Let

$$\begin{aligned} f :&\mathbb{R}^{3N} \times \mathbb{R} \to \mathbb{R}^{3N} \\ &(\mathbf{x}_i, t) \mapsto f(\mathbf{x}_i, t) \end{aligned} \tag{3.15}$$

denote the *deterministic* part of the set of model equations (5.1), while

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_u, \boldsymbol{\beta}_h, \boldsymbol{\beta}_r) \in \mathbb{R}^{3N} \tag{3.16}$$

is the *stochastic* part of the model.

Truth model

For a time index $j \in \mathbb{N}^*$, the *unmodified model* can be written as

$$\mathbf{x}^j = f(\mathbf{x}^{j-1}) + \boldsymbol{\beta}^j, \tag{3.17}$$

where $\boldsymbol{\beta}^j \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$, is the Gaussian model error at time j with covariance matrix $\mathbf{Q} \in \mathbb{R}^{3N \times 3N}$. The distribution function of $\boldsymbol{\beta}^j \in \mathbb{R}^{3N}$ is given by p in (2.34).

Model of the ensemble

An ensemble of k particles is integrated forward in time and generated according to the importance density q, which is chosen such that the particles are nudged, at each integration time step, towards the new incoming observations. The modified model equation that governs the ensemble is

$$\mathbf{x}_i^j = f(\mathbf{x}_i^{j-1}) + \hat{\boldsymbol{\beta}}_i^j + \mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})), \tag{3.18}$$

where $\mathbf{d}^n$ is the $3m$-dimensional, future observation vector at time $n \geq j$. $\mathbf{K} \in \mathbb{R}^{3N \times 3m}$ is the nudging matrix that determines the magnitude of relaxation by which the particles get closer to the observations. $\mathbf{H} \in \mathbb{R}^{3m \times 3N}$ is the observation operator that converts the dimension of state vectors to the dimension of observations, $m \leq N$. The error of the above model (3.18) for the i-th particle at time j is

$$\hat{\boldsymbol{\beta}}_i^j + \mathbf{a}_i^j, \quad \mathbf{a}_i^j = \mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})), \tag{3.19}$$

where $\hat{\boldsymbol{\beta}}_i^j \sim \mathcal{N}(0, \hat{\mathbf{Q}})$. This model error by which erroneously modified trajectories of particles are generated has the Gaussian probability distribution function q, with mean $\mathbf{a}_i^j \in \mathbb{R}^{3N}$ and covariance matrix $\hat{\mathbf{Q}} \in \mathbb{R}^{\mathbf{3N \times 3N}}$.

Model of the observation errors

For the following experiments, we choose the observation errors to have a Gaussian distribution as well so that

$$\mathbf{d}^n - \mathbf{H}\mathbf{x}_i^n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{3.20}$$

where $\mathbf{R} \in \mathbb{R}^{3m \times 3m}$ is the observation error covariance matrix.

Weights

Assume that new observations arrive after every L-th integration time step. The p-density, the q-density and the observation error density above enable us to compute the weights according to (2.49),

$$w_i = p(\mathbf{d}^n|\mathbf{x}_i^n)\frac{p(\mathbf{x}_i^n|\mathbf{x}_i^{n-L})}{q(\mathbf{x}_i^n|\mathbf{x}_i^{n-L},\mathbf{d}^n)},$$

namely as

$$\begin{aligned} w_i \sim exp\big(-\frac{1}{2}&\underbrace{(\mathbf{d}^n-\mathbf{H}\mathbf{x}_i^j)^T\mathbf{R}^{-1}(\mathbf{d}^n-\mathbf{H}\mathbf{x}_i^j)}_{1}- \\ -\frac{1}{2}&\underbrace{\sum_{j=n-L+1}^{n}(\mathbf{K}(\mathbf{d}^n-\mathbf{H}f(\mathbf{x}_i^{j-1}))+\hat{\boldsymbol{\beta}}_i^j)^T\mathbf{Q}^{-1}(K(\mathbf{d}^n-\mathbf{H}f(\mathbf{x}_i^{j-1}))+\hat{\boldsymbol{\beta}}_i^j)}_{2} \\ +\frac{1}{2}&\underbrace{\sum_{j=n-L+1}^{n}(\hat{\boldsymbol{\beta}}_i^j)^T\hat{\mathbf{Q}}^{-1}\hat{\boldsymbol{\beta}}_i^j}_{3}\big). \end{aligned} \tag{3.21}$$

The proportionality constant factor of the exponent can be neglected in what follows, due to normalization of the weights which makes this constant, common to all weights, irrelevant for their computation.

3.2.2 Weight equalization and resampling

Testing the filter with nudging but without equalizing the weights at the last time step shows that the filter collapses, after one or two assimilation time steps, to at most two ensemble members. Therefore, we implemented the equalization procedure following [33], [1], in a slightly modified way.

At each small integration time step (all but the first one, as mentioned in Chapter 2), the particles are generated from the q-density,

$$\mathbf{x}_i^n = f(\mathbf{x}_i^{n-1}) + \hat{\boldsymbol{\beta}}_i^n + \mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})). \tag{3.22}$$

At the very last integration time step, before resampling starts a new assimilation cycle, the particles are nudged separately in such a way that their weights become equal. To that purpose, an additional real parameter α is introduced at the last time step in the model equation

$$\mathbf{x}_i^n = f(\mathbf{x}_i^{n-1}) + \alpha_i\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})), \tag{3.23}$$

while leaving out for now the stochastic perturbation $\boldsymbol{\beta}_i^n$. The last step stochastic perturbation $\boldsymbol{\beta}_i^n$ will be re-added in the weights and the model after calculating the α_i-parameter that ensures equal weights. Replacing $\mathbf{x}_i^n$ by its value depending on α_i, the weights become

$$
\begin{aligned}
w_i \sim \exp\big(&-\frac{1}{2}(\mathbf{d}^n - \mathbf{H}\mathbf{x}_i^n)^T\mathbf{R}^{-1}(\mathbf{d}^n - \mathbf{H}\mathbf{x}_i^n) - \\
-\frac{1}{2}\sum_{j=n-L+1}^{n} &(\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})) + \hat{\boldsymbol{\beta}}_i^n)^T\mathbf{Q}^{-1}(\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})) + \hat{\boldsymbol{\beta}}_i^j) \\
&+\frac{1}{2}\sum_{j=n-L+1}^{n}(\hat{\boldsymbol{\beta}}_i^j)^T\hat{\mathbf{Q}}^{-1}\hat{\boldsymbol{\beta}}_i^j) \\
= \exp\big(&-\frac{1}{2}(\mathbf{d}^n - \mathbf{H}(f(\mathbf{x}_i^{n-1}) + \alpha_i\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1}))))^T\mathbf{R}^{-1} \\
&\times (\mathbf{d}^n - \mathbf{H}(f(\mathbf{x}_i^{n-1}) + \alpha_i\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})))) \\
-\frac{1}{2}\sum_{j=n-L+1}^{n-1} &(\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})) + \hat{\boldsymbol{\beta}}_i^j)^T\mathbf{Q}^{-1}(\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})) + \hat{\boldsymbol{\beta}}_i^j) \\
&-\frac{1}{2}(\alpha_i\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})))^T\mathbf{Q}^{-1}(\alpha_i\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1}))) \\
&+\frac{1}{2}\sum_{j=n-L+1}^{n-1}(\hat{\boldsymbol{\beta}}_i^j)^T\hat{\mathbf{Q}}^{-1}\hat{\boldsymbol{\beta}}_i^j),
\end{aligned}
\tag{3.24}
$$

Then, using the abbreviation $\boldsymbol{\delta}_i^j := \mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{j-1})$, the weights can be reformulated after some simplifying manipulations as

$$
w_i \sim \exp\big(- A_i\alpha_i^2 + B_i\alpha_i - C_i), \tag{3.25}
$$

where

$$
A_i = \frac{1}{2}(\mathbf{K}\boldsymbol{\delta}_i^n)^T(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{Q}^{-1})(\mathbf{K}\boldsymbol{\delta}_i^n) \tag{3.26}
$$

$$
B_i = (\boldsymbol{\delta}_i^n)^T\mathbf{R}^{-1}\mathbf{H}\mathbf{K}\boldsymbol{\delta}_i^n \tag{3.27}
$$

$$
\begin{aligned}
C_i = &\frac{1}{2}(\boldsymbol{\delta}_i^n)^T\mathbf{R}^{-1}(\boldsymbol{\delta}_i^n) + \frac{1}{2}\sum_{j=n-L+1}^{n-1}(\boldsymbol{\beta}_i^j + \mathbf{K}\boldsymbol{\delta}_\mathbf{i}^\mathbf{j})\mathbf{Q}^{\mathbf{-1}}(\boldsymbol{\beta}_\mathbf{i}^\mathbf{j} + \mathbf{K}\boldsymbol{\delta}_\mathbf{i}^\mathbf{j}) \\
&-\sum_{j=n-L+1}^{n-1}(\boldsymbol{\beta}_i^j)^T\hat{\mathbf{Q}}^{\mathbf{-1}}(\boldsymbol{\beta}_\mathbf{i}^\mathbf{j}).
\end{aligned}
\tag{3.28}
$$

When $\mathbf{H} = \mathbf{Id}$, the constants A_i and B_i simplify to

$$
A_i = \frac{1}{2}(\mathbf{K}\boldsymbol{\delta}_i^n)^T(\mathbf{R}^{-1} + \mathbf{Q}^{-1})(\mathbf{K}\boldsymbol{\delta}_i^n) \tag{3.29}
$$

$$
B_i = (\boldsymbol{\delta}_i^n)^T\mathbf{R}^{-1}\mathbf{K}\boldsymbol{\delta}_i^n. \tag{3.30}
$$

Finding the maximum weight for each particle

By adding this real-valued parameter α_i for each particle to the model equation at the last time step just before resampling, the different weights can be maximized individually with respect to the corresponding parameter by solving the second order equation

$$\begin{aligned} & -A_i\alpha_i^2 + B_i\alpha_i - C_i \to \max \\ \Leftrightarrow \quad & f(\alpha_i) := A_i\alpha_i^2 - B_i\alpha_i + C_i \to \min. \end{aligned}$$

The minimum for f is obtained for

$$(\alpha_i)_{\min} = \frac{B_i}{2A_i}. \tag{3.31}$$

It corresponds to the maximum value that the weight of particle i can achieve by replacing the nudging matrix $\mathbf{K}$ by $\alpha_i\mathbf{K}$ in the last time step.

Setting almost all weights equal to a target weight

For every particle i, the function $f(\alpha_i) = -\log w(\alpha_i)$ describes a quadratic function with attains its minimum at $(\alpha_i)_{\min}$. In order to be able to resample many particles, it is necessary that the weights have similar magnitude. Therefore, following [33] and [1] we choose a target weight value which all or at least an arbitrary percentage number of the weights can reach by nudging the particles with corresponding α at the last time step. This is done numerically by ordering the maxima $w((\alpha_i)_{\min})$ of the weights in decreasing order and then determining the weight max for which a pre-determined percentage of particles can reach it. We follow [33] and choose 80 percent of the particles to be equal.

Once the target weight is determined, the parameter α_i is computed for each particle such that this particle is nudged to have its weight reach the target weight. This is done according to

$$\begin{aligned} w_i &= (w_i)_{\text{target}} \\ \Leftrightarrow -A_i\alpha_i^2 + B_i\alpha_i - C_i &= \log(w_i)_{\text{target}} \\ \Leftrightarrow \alpha_i^2 - \frac{B_i}{A_i}\alpha_i + \frac{C_i - \log(w_i)_{\text{target}}}{A_i} &= 0. \end{aligned} \tag{3.32}$$

We solve this quadratic equation for each i and take the positive solution

$$(\alpha_i)_{\text{target}} = \frac{B_i}{2A_i} + \sqrt{\left(\frac{B_i}{2A_i}\right)^2 - \frac{C_i - \log(w_i)_{\text{target}}}{A_i}} \tag{3.33}$$

Re-generating the particles and re-adding a stochastic term to the model equation at the last time step

Now, the particles can be nudged at the last time step following the model equation

$$\mathbf{x}_i^n = f(\mathbf{x}^{n-1})_i + (\alpha_i)_{\text{target}}\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})) + \boldsymbol{\beta}_i^n, \tag{3.34}$$

where $\boldsymbol{\beta}_i^n$ is the stochastic perturbation at the last integration time step that was cast out for the computation of the target weight. It is necessary to introduce it again, for otherwise the density q would be a delta function, reduced to zero for all states that are not equal to the generated vector and thus inducing a division by zero, which is not allowed and not defined [33]. We generate this stochastic perturbation following [1] with a *mixture density* [2] in such a way that the equalized weights keep very similar and do not change much by the perturbation of the particles. This is done by choosing the Importance density q at the last time step to be

$$q = (1-\epsilon)\mathcal{U}(0,\gamma) + \epsilon\mathcal{N}(0,\gamma\mathbf{I}), \tag{3.35}$$

where

$$\gamma = 10^{-4} \quad \text{and} \quad \epsilon = \frac{1}{1000}k^{-1} \tag{3.36}$$

are very small numbers, and where k is the ensemble size. This choice of such a density is based on the wish to introduce a density with a small amplitude in order not to change the particles too much, and with a large width so that the weights do not change much[33]. The perturbation vector $\boldsymbol{\beta}$ is generated by picking a value $\lambda \sim \mathcal{U}[0,1]$. Then,

$$\boldsymbol{\beta}_i^n \sim \begin{cases} \mathcal{N}(0,\gamma^2\mathbf{I}) & \text{if} \quad \lambda < \epsilon \\ \mathcal{U}(0,\gamma) & \quad \text{else.} \end{cases} \tag{3.37}$$

Since ϵ is so small, it is most probable that the random vector $\boldsymbol{\beta}_i^n$ will be drawn from the uniform distribution, and very unlikely that it is drawn from the Gaussian distribution. But still, as pointed out in [1], the existence of the Gaussian density ensures the continuous support of the distribution and the fact that non zero random perturbation vectors are generated on the whole vector space.

Once the particles are re-generated with a stochastic term, their model error contribution to the weights at the last time step can be computed before resampling, and the weight of particle i at time resampling time step n becomes

$$w_i^n = w_i^{n-1}\frac{p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1})}{q(\mathbf{x}_i^n|\mathbf{x}_i^{n-1},\mathbf{d}^n)}, \tag{3.38}$$

where

$$p(\mathbf{x}_i^n|\mathbf{x}_i^{n-1}) \sim \exp\Bigg(-\frac{1}{2}((\alpha_i)_{\text{target}}\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})) + \hat{\boldsymbol{\beta}}_i^n)^T\mathbf{Q}^{-1}\times$$
$$((\alpha_i)_{\text{target}}K(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}_i^{n-1})) + \hat{\boldsymbol{\beta}}_i^n)\Bigg),$$

[2] [1] between a uniform and a Gaussian distribution, Section 3.2.4

and

$$q = \frac{1-\epsilon}{(2\gamma)^N} \left(1_{[-\gamma,\gamma]} + \frac{\epsilon}{1-\epsilon} \left(\frac{2}{\pi} \right)^{\frac{N}{2}} \gamma^{N-1} \right) \exp^{-\frac{1}{2}(\boldsymbol{\beta}_i^n)^T (\gamma \mathbf{I})^{-1} \boldsymbol{\beta}_i^n} . \tag{3.39}$$

Chapter 4

Testing the Efficient Particle Filter on the stochastic cloud model

This chapter presents the results of testing the efficient particle filter on the stochastic cloud model [58]. This model is the first one in a hierarchy of testbed models that were developed for convective scale data assimilation (cf. Section 3). Though very simple in its design, it corresponds to the minimal requirements for the simulation of convection. It is purely stochastic and highly nonlinear, and thus represents an extreme testbed for particle filters, which were especially designed to handle dynamics with sudden and rapid changes.

The method used for data assimilation is a simplified version of the efficient particle filter [33] in which the equalization of weights is left out, in order to understand basic features of the filter. A special focus in this section is given to the shape of the weights and to the selection mechanism that takes place in the frame of resampling. To this purpose, we derive the exact probability density function of the stochastic birth-death process in section 4.1.1. In Section 4.1.2, we analyze the consequences of this probability density on the filter behaviour. Finally, the test settings and the results of our particle filter experiments are presented in Sections 4.2 and 4.3.

4.1 Efficient particle filter without weight equalization

4.1.1 Deriving the weights for a non-Gaussian model

The model variable at time t, denoted by $\boldsymbol{\psi}^t$ ($\boldsymbol{\psi}^t \in \mathbb{N}^N$), is an N-tuple of integer valued numbers of clouds, and it represents the number of existing clouds at each grid point. Let us omit for simplicity, and just for now, the index $i \in [1, k]$ of the particles. The model evolution is governed by the birth-death-process f,

$$\boldsymbol{\psi}^t = f(\boldsymbol{\psi}^{t-1}), \tag{4.1}$$

where f is the stochastic model and t the discrete time index. The stochastic nature of the model is entirely determined by the transition probability density function $p(\boldsymbol{\psi}^t|\boldsymbol{\psi}^{t-1})$, which will be derived in the next subsection.

In the following assimilation setup, the importance density q is chosen as in [33]. The model equation is modified by artificially nudging the state vectors towards the future observations. After an initial draw from the prior, given by the Poisson distribution (3.1), instead of time-integrating particles according to p (i.e. following (4.1)), they are generated following q, described by the modified model equation

$$\boldsymbol{\psi}^t = f(\boldsymbol{\psi}^{t-1}) + \mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\boldsymbol{\psi}^{t-1})), \tag{4.2}$$

where $\mathbf{K} \in \mathbb{R}^{\mathbf{N}\times\mathbf{m}}$ is the relaxation or nudging matrix, N is the dimension of the state vector and $m \leq N$ the dimension of the observation vector. In this way, particles are nudged towards the new incoming, future observations ($\mathbf{d}^n$) at time n, and the matrix $\mathbf{K}$ determines by which factor the distance between integrated state vector and observations is reduced in order to approach the observation density by the prior. For this model, the observations are chosen to be available at every grid point, hence $m = N$ and the observation operator $\mathbf{H} \in \mathbb{R}^{\mathbf{m}\times\mathbf{N}}$, becomes $\mathbf{H} = \mathbf{Id_N}$. Therefore, $\mathbf{H}$ will be omitted.. Similarly, $\mathbf{K} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}}$ is now a symmetric and diagonal matrix.

Defining the events in probabilistic space

Consider a single grid point $l \in \mathbb{N}$ with a existing clouds at time $t-1$, ($t \in \mathbb{N}$). At time t, either one single cloud is added with probability λ, or each of the existing a clouds independently from the others dies with probability μ. A possible 'event' happening at this grid point therefore is, for example, the $(a+1)$-dimensional vector $\mathbf{x} \in (\{\mathbf{0},\mathbf{1}\})^{\mathbf{a+1}}$ defined on the space of binary events, given by

$$\mathbf{x} := (1, 0, ..., 0),$$

where $x_0 = 1$ stands for 'a cloud appears' and $x_0 = 0$ for 'no cloud appears', and where for each $j \in [1, a]$, $x_j = 0$ is equivalent to 'cloud j dies' and $x_j = 1$ to 'cloud j remains', its contrary. For this event, we have $p(x_0) = \lambda$, $p(x_1) = \mu$,..., $p(x_a) = \mu$, and what happens at each gridpoint is independent of the other grid points.

The resulting number b of clouds at time t for this event is equal to 1 and takes place with probability $p(b = 1) = \lambda\mu^a$. Thus, one can define the family of 2^{a+1} such possible 'events' by

$$\{\mathbf{x} \in \mathbb{N}^{a+1} | \forall j \in [0, a+1], x_j \in \{0, 1\}\}.$$

Let ψ^t be the integer number of clouds at time t that results from such an event at a fixed grid point. This result at time t depends on the result at time $t-1$. Consequently, it is possible to compute the transition probability density function of ψ from time $t-1$ to time t.

Transition probability density function $p(\boldsymbol{\psi}^t|\boldsymbol{\psi}^{t-1})$

For the following, consider a one-dimensional particle ψ^t. Let us set $\psi^t =: b$ and $\psi^{t-1} =: a$, $(a, b \in \mathbb{N})$. The characteristic function of a distribution p of a discrete random variable a can be expressed[1] as

$$\phi_a(y) = \sum_{b=0}^{\infty} p(b|a)e^{iyb}, \tag{4.3}$$

where y is a parameter. The characteristic function entirely determines the distribution function p. The design of the model imposes that, since only one additional cloud is potentially born,

$$\forall b > a + 1, \quad p(b|a) = 0 \tag{4.4}$$

for a given $a \in \mathbb{N}$. If there is $a = 1$ cloud at at time $t-1$ at given grid point, there can only be 0,1 or 2 clouds at consecutive time t. This means that the possible transition densities can only be

$$p(b|a = 1) \in \{p(0|1), p(1|1), p(2|1)\}$$

Equivalently,

$$p(b|1) = \{(1 - \lambda)\mu, \lambda\mu, \lambda(1 - \mu)\}$$

Let us set $x(y) := x = e^{iy}$ to simplify the notation in what follows. y is the parameter of the Fourier transform $\phi(y) = E[e^{iby}]$ of the random variable b. It follows from (4.3) that

$$\phi_1(x) = p(0|1)x^0 + p(1|1)x^1 + p(2|1)x^2,$$

and hence,

$$\phi_1(x) = (1 - \lambda)\mu + [\lambda\mu + (1 - \lambda)(1 - \mu)]x + \lambda(1 - \mu)x^2.$$

Replacing the former number of clouds $\psi^{t-1} = a = 1$ by $a > 1$, it can be shown by induction that

$$\forall a \in \mathbb{N}^*, \phi_a(x) = (\lambda x + (1 - \lambda))(\mu + (1 - \mu)x)^a. \tag{4.5}$$

Proof.

- $a = 0$: If there is no cloud at time $t - 1$, the design of the stochastic model permits only birth, or not, of another cloud. Since no cloud exists at time $t - 1$, no cloud can die. Hence, we have

  $$p(n|0) \in \{p(0|0), p(1|0)\}, \tag{4.6}$$

  with

  $$p(0|0) = (1 - \lambda), \quad p(1|0) = \lambda. \tag{4.7}$$

  Hence

  $$\phi_0(x) = \lambda x + (1 - \lambda). \tag{4.8}$$

[1] see for example [27].

- a-1 → a: Suppose that we have a existing clouds at time $t-1$, and that
$$\phi_{a-1}(x) = (\lambda x + (1-\lambda))(\mu + (1-\mu)x)^{a-1} \tag{4.9}$$
is true. Then, the resulting number of clouds at time t can be $t \in \{0, 1, ..., a+1\}$. We compute $p(0|a), p(1|a), ..., p(a+1|a)$.
  - $p(0|a)$: Starting from a clouds, there is just one way to end up with no clouds: no cloud appears, and all existing clouds disappear. Therefore,
$$p(0|a) = (1-\lambda)\mu^a. \tag{4.10}$$
  - $p(1|a)$: In order to end up with 1 cloud at time t, there are more options. Either an additional cloud is born and all a existing clouds die, or no cloud is born and all but one clouds die, the surviving cloud having $\binom{a}{1}$ possible choices to be selected out of the a ones. We have
$$p(1|a) = (1-\lambda)(1-\mu)\mu^{a-1} + \binom{a}{1}\mu^{a-1}(1-\mu). \tag{4.11}$$

    In the same way, we find
  - $p(b|a)\forall b \in [2, a]$:
$$\begin{aligned} p(2|a) &= (1-\lambda)\binom{a}{2}\mu^{a-2}(1-\mu)^2 + \lambda\binom{a}{1}\mu^{a-1}(1-\mu) \\ &\;\;... \\ p(a|a) &= (1-\lambda)\binom{a}{a}(1-\mu)^a + \lambda\binom{a}{a-1}\mu(1-\mu)^{a-1} \end{aligned} \tag{4.12}$$

    And finally,
$$p(a+1|a) = \lambda(1-\mu)^{a+1}. \tag{4.13}$$

The above transition probability densities are now introduced into the characteristic function (4.3). We have
$$\begin{aligned} \phi_a(x) = (1-\lambda)\mu^a x^0 + \\ &+ [(1-\lambda)(1-\mu)\mu^{a-1} + \binom{a}{1}\mu^{a-1}(1-\mu)]x^1 \\ &+ [(1-\lambda)\binom{a}{2}\mu^{a-2}(1-\mu)^2 + \lambda\binom{a}{1}\mu^{a-1}(1-\mu)]x^2 \\ &+ ... \\ &+ [(1-\lambda)\binom{a}{a}(1-\mu)^a + \lambda\binom{a}{a-1}\mu(1-\mu)^{a-1}]x^a \\ &+ [\lambda(1-\mu)^{a+1}]x^{a+1}, \end{aligned} \tag{4.14}$$

which can be regrouped into

$$\begin{aligned}\phi_a(x) &= (\lambda x + (1-\lambda)) \sum_{b=0}^{a} \binom{a}{b} \mu^b (1-\mu)^{a-b} x^{a-b} \\ &= (\lambda x + (1-\lambda)) \left(\mu + (1-\mu)x\right)^a \\ &= \phi_{a-1}(x)(\mu + (1-\mu)x), \end{aligned} \tag{4.15}$$

where we used the binomial equation. $\phi_{a-1}(x)$ is the characteristic function for $a-1$ clouds at time $t-1$, by the induction hypothesis (4.9).

□

Expressing the binomial sum in (4.5), we finally have

$$\begin{aligned}\phi_a(x) &= (\lambda x + (1-\lambda))(\mu + (1-\mu)x)^a \\ &= \sum_{b=0}^{a} \binom{a}{b} \mu^b (1-\mu)^{a-b} [\lambda x^{a-b+1} + (1-\lambda)x^{a-b}]. \end{aligned} \tag{4.16}$$

The coefficients c^b of x^b in the polynomial expansion of ϕ in (4.3) are the transition probability densities $p(b|a)$.

After a comparison of coefficients of the same order between (4.3) and (4.16), we get:

$$\forall (b,a) \in \mathbb{N} \times \mathbb{N}, \quad p(b|a) = \tag{4.17}$$

$$\begin{cases} \mu^a (1-\lambda) & \text{if } b = 0 \\ \mu^{a-b}(1-\mu)^{b-1} \left(\binom{a}{a-b+1} \lambda\mu + \binom{a}{a-b} (1-\lambda)(1-\mu) \right) & \text{if } 1 \leq b \leq a \\ (1-\mu)^a \lambda & \text{if } b = a+1 \end{cases}$$

This is the expression of the transition probability density function from time $t-1$ to t of one random variable 'number of clouds' (i.e. one particle) on a single grid point. The derivation of the transition density for N-dimensional particles is straightforward. Since the realizations of cloud numbers are independent from each other on each grid point, the characteristic function in the multidimensional case becomes

$$\phi_{a_1, a_2, \ldots, a_N}(x_1, \ldots, x_N) = \prod_{l=0}^{N} \phi_{a_l}(x_l)$$

for N gridpoints. For the same reason of independency with respect to different grid points, the multidimensional probability density functions are given by

$$p(b_1, b_2, \ldots, b_N | a_1, \ldots, a_N) = \prod_{l=0}^{N} p(b_l | a_l),$$

where the densities $p(b_l|a_l)$ are given by (4.17) for each grid point l separately.

Weights for the stochastic cloud model

For expressing the weights, let us introduce the index i, $i \in [1, ..., k]$, to characterize k particles of dimension N, and re-define these particles as

$$\begin{aligned} \boldsymbol{\psi}_i^t &: = (b_1^i, b_2^i, ..., b_N^i), \\ \boldsymbol{\psi}_i^{t-1} &:= (a_1^i, a_2^i, ..., a_N^i). \end{aligned} \tag{4.18}$$

Then,

$$p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}) = \prod_{l=0}^{N} p(\psi_{i,l}^t|\psi_{i,l}^{t-1}) = \prod_{l=0}^{N} p(b_l^i|a_l^i), , \tag{4.19}$$

where $p(n_l^i|m_l^i)$ is given by the expression (4.17) for each particle $i \in [1, k]$ and each grid point $l \in [1, N]$ separately.

In order to distinguish the two state vectors generated from p and q respectively, let us for now introduce $\boldsymbol{\phi}_i^t$ and replace $\boldsymbol{\psi}_i^t$ by $\boldsymbol{\phi}_i^t$ in the equation (4.1). The error density $p(\boldsymbol{\phi}_i^t|\boldsymbol{\psi}_i^{t-1})$ of the unmodified model is entirely determined by the stochastic birth-death process f

$$\boldsymbol{\phi}_i^t = f(\boldsymbol{\psi}_i^{t-1}), \tag{4.20}$$

while the model error density $q(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}, \mathbf{d}^n)$ of the nudged equation is defined through the equation (4.2) with nudging,

$$\boldsymbol{\psi}_i^t = f(\boldsymbol{\psi}_i^{t-1}) + \mathbf{K}(\mathbf{d}^n - f(\boldsymbol{\psi}_i^{t-1})). \tag{4.21}$$

But since the only stochastic source of equation (4.21) is f as well as of equation (4.20), once the transition density p is known, the proposal density q can be derived from p by a linear transformation of $\boldsymbol{\psi}$. Substituting $\boldsymbol{\phi}_i^t$ into equation (4.21), we have

$$\boldsymbol{\psi}_i^t = \boldsymbol{\phi}_i^t + \mathbf{K}(\mathbf{d}^n - \boldsymbol{\phi}_i^t), \tag{4.22}$$

whence $\boldsymbol{\phi}_i^t$ can be expressed as a function of $\boldsymbol{\psi}_i^t$,

$$\boldsymbol{\phi}_i^t = (\mathbf{Id} - \mathbf{K})^{-1}(\boldsymbol{\psi}_i^t - \mathbf{K}\mathbf{d}^n). \tag{4.23}$$

In case the model state was not required to be integer-valued, there would be a one-to one relationship between $\boldsymbol{\psi}_i^t$ and $\boldsymbol{\phi}_i^t$ provided that $\mathbf{Id} - \mathbf{KH}$ is invertible, in which case $\boldsymbol{\psi}_i^t$ could have a certain value if and only if $\boldsymbol{\phi}_i^t$ took that value. However, since this model is discrete, it imposes to arbitrarily replace the value of ϕ by an integer. We decided to round ϕ to the nearest integer, which makes sense with regard to nudging. Denoting by $\{x\}, x \in \mathbb{R}$, the rounded number to the nearest integer, and consider one grid point, we can write

$$\{x\} = \lfloor x + 0.5 \rfloor, \tag{4.24}$$

where '$\lfloor$' is the smallest and nearest integer to which a real value is rounded. From now on, let $\boldsymbol{\phi}_i^t := \{\boldsymbol{\phi}_i^t\}$. Then (4.23) becomes

$$\boldsymbol{\phi}_i^t = \{(\mathbf{Id} - \mathbf{K})^{-1}(\boldsymbol{\psi}_i^t - \mathbf{K}\mathbf{d}^n)\}. \tag{4.25}$$

Thus, the Importance density q differs from p only by a scaling factor and is determined by

$$\begin{aligned} q(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}, \mathbf{d}^n) &= p(\boldsymbol{\phi}_i^t|\boldsymbol{\psi}_i^{t-1}) &(4.26)\\ &= p(\{(\mathbf{Id} - \mathbf{K})^{-1}(\boldsymbol{\psi}_i^t - \mathbf{K}\mathbf{d}^n)\}|\boldsymbol{\psi}_i^{t-1}). &(4.27)\end{aligned}$$

Once we know the density function of the observations $\mathbf{d}^n$ and the transition density function p (and therefore q), the weights of the particles can be computed following the Importance resampling method with nudging [33]. In accordance with what precedes, they are given by

$$\begin{aligned} w_i^t &= p(\mathbf{d}^n|\boldsymbol{\psi}_i^t)\frac{p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1})}{q(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}, \mathbf{d}^n)}, \\ &= p(\mathbf{d}^n|\boldsymbol{\psi}_i^t)\frac{p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1})}{p(\boldsymbol{\phi}_i^t|\boldsymbol{\psi}_i^{t-1})}, &(4.28)\end{aligned}$$

where i is the i-th particle, t the time index and n the time step of the future observation, and where the density p is substituted with the expression in (4.19). This means that at every time step t of the assimilation experiment undertaken with the efficient particle filter, the weight w_i^t of each ensemble member i is computed according to the above expression (4.28).

Computational procedure

In practice, and to summarize, for each particle i and at time t, one would

- Generate particle $\boldsymbol{\psi}_i^t$ according to q:

$$\boldsymbol{\psi}_i^t = f(\boldsymbol{\psi}_i^{t-1}) + \mathbf{K}(\mathbf{d}^n - f(\boldsymbol{\psi}_i^{t-1})). \tag{4.29}$$

  Make particles integer valued: $\boldsymbol{\psi}_i^t \to \boldsymbol{\psi}_i^t := \{\boldsymbol{\psi}_i^t\}$ before evaluating their density.

- Evaluate $p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1})$:

$$p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}) = \prod_{l=0}^{N} p(\psi_{i,l}^t|\psi_{i,l}^{t-1}), \tag{4.30}$$

  where

$$p(\psi_{i,l}^t|\psi_{i,l}^{t-1}) = \tag{4.31}$$

$$\begin{cases} \mu^{\psi_{i,l}^{t-1}}(1-\lambda) & \text{if } \psi_{i,l}^t = 0 \\ \mu^{\psi_{i,l}^{t-1}-\psi_{i,l}^t}(1-\mu)^{\psi_{i,l}^t-1}\left(\binom{\psi_{i,l}^{t-1}}{\psi_{i,l}^{t-1}-\psi_{i,l}^t+1}\lambda\mu + \binom{\psi_{i,l}^{t-1}}{\boldsymbol{\psi}_{i,l}^{t-1}-\psi_{i,l}^t}(1-\lambda)(1-\mu)\right) & \text{if } 1 \leq \psi_{i,l}^t \leq \psi_{i,l}^{t-1} \\ (1-\mu)^{\psi_{i,l}^{t-1}}\lambda & \text{if } \psi_{i,l}^t \geq \psi_{i,l}^{t-1} + 1. \end{cases}$$

- Evaluate $q(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}, \mathbf{d}^n)$ according to

$$q(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}, \mathbf{d}^n) = p(\{(\mathbf{Id} - \mathbf{K})^{-1}(\boldsymbol{\psi}_i^t - \mathbf{K}\mathbf{d}^n)\}|\boldsymbol{\psi}_i^{t-1}) \tag{4.32}$$

- Evaluate $p(\mathbf{d}^n|\boldsymbol{\psi}_i^t)$: we chose (cf. Section 4.2) either an *exponential* distribution of the observation errors

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n)_{\text{exp}} = C_1 \exp\left(-\frac{\sqrt{(\boldsymbol{\psi}_i^n - \mathbf{d}^n)^2}}{\sigma_{\text{e}}}\right), \tag{4.33}$$

  where $\mathbf{d}^n$ is the observation at time n and σ_{e} the parameter of the exponential distribution, or a *Gaussian* one,

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n)_{\text{gauss}} = C_2 \exp\left(-\frac{(\boldsymbol{\psi}_i^n - \mathbf{d}^n)^2}{2\sigma_{\text{g}}^2}\right), \tag{4.34}$$

  where σ_{g}^2 is the observation error variance and where C_1 and C_2 are the normalization constants. The latter have no impact on the value of the weights since they are common to all particles and hence are factorized out during the normalization step of the weights.

- Compute the weights

$$w_i^t = p(\mathbf{d}^n|\boldsymbol{\psi}_i^n)\frac{p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1})}{p(\boldsymbol{\phi}_i^t|\boldsymbol{\psi}_i^{t-1})}, \quad \boldsymbol{\phi}_i^t = \{(\mathbf{Id} - \mathbf{K})^{-1}(\boldsymbol{\psi}_i^t - \mathbf{K}\mathbf{d}^n)\}, \tag{4.35}$$

  where $\boldsymbol{\phi}_i^t$ is the integer value of a merely rescaled state vector $\boldsymbol{\psi}_i^t$.

Remark

We replaced the equality $\psi^t = \psi^{t-1} + 1$ in the conditioning of (4.17) by an inequality in (4.32), $\psi^t \geq \psi^{t-1} + 1$. This is done in order to take into account the case where *more than one cloud appear* at a grid point, in spite of the given birth-death process which strictly speaking only allows the birth of one cloud. But since the model undergoes a stochastic perturbation after each resampling step (cf. [58]), and also in case the cloud density would be increased, it is possible indeed that more than just one cloud appear at a grid point.

This case, however, remains highly unprobable and needs to be 'penalized' by a small value of corresponding probability. We thus choose, for this probability value, the first coefficient of order larger or equal than $\psi^{t-1}+1$ which would appear in the series expansion of the *Poisson birth process* process that allows as many clouds to be born as to die,

$$\phi_{\psi^{t-1}}(y) = \sum_{\psi^t=0}^{\psi^{t-1}} p(\psi^t|\psi^{t-1})e^{iy\psi^t} + \sum_{\psi^t=\psi^{t-1}+1}^{\infty} p(\psi^t|\psi^{t-1})e^{iy\psi^t}. \tag{4.36}$$

This coefficient corresponds to the largest value of the remaining sum on the right in (4.36) that would appear within the characteristic function if birth of more clouds were allowed. It is the largest probability that can arise from this unlikely situation of birth of more than one cloud, and it is given by

$$(1-\mu)^{\psi_{i,l}^{t-1}}\lambda. \tag{4.37}$$

4.1.2 Consequences on the filter

A competition between two types of densities

The weights are determinant for the selection of the particles during the resampling procedure, which replicates high-weighted particles and discards those with low weights. The shape of the densities that are involved in the weights' computation, given by (4.35), and in particluar the combination of the observation density and the p- and q- densities, has a singular effect on the behaviour of the filter (cf. Results in next Section). The p-density and hence the q-density represent the probability of the *dynamical evolution* of a state (an N-tuple of numbers of clouds) to another, while the observation density is the probability of a *static* position of the observation, given the present state vector. Both types of probabilities, the static observation density and the p-/q-density, are driven by different laws, which are both directly impacted by nudging.

The observation density (exponential or Gaussian, cf.(4.33) and (4.34)) is determined by its parameter and by the distance between model and observation at the same present time step. If model and observation are very far from each other, the observation density becomes very small due to the negative exponent which is increased by a small density parameter (eg $\sigma_e = 0.05$). If on the contrary, model and observation coincide (either by chance or more probably through strong nudging), the probability density function of the observation error is equal to 1.0, whereby the weight is then only determined by the p/q-density fraction. For an exponential observation error density with paramater of $\sigma_e = 0.05$ and $\Delta := \sqrt{(\boldsymbol{\psi}_i^n - \mathbf{d}^n)^2}$, we have either one of the two cases,

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) \sim \begin{cases} \lesssim e^{-20\Delta} \leq e^{-20} \approx 2.10^{-9} & \text{if } \Delta \neq 0 \\ = 1.0 & \text{if } \Delta = 0. \end{cases} \tag{4.38}$$

For the case of a Gaussian observation density, the same approximation as (4.38) holds, with the difference that Δ is replaced by Δ^2 and that therefore, the probability density function converges much more rapidly to zero with increasing distance Δ, than in the case of an exponential observation density.

As to the p-density and hence the q-density, they essentially depend on the birth and death probabilities, consequently on the mean cloud density per grid point and the cloud half life through expression (3.3). If the expected life time of a cloud is high ($hl = 3000$), resulting in a birth and death probability of $\lambda \sim 2.10^{-5}$ and $\mu \sim 2.10^{-4}$ respectively, then expression (4.31) shows that *a change will occur with low probability* while *a stationary state will have high probability*, close to 1. It suffices that one single cloud arises or disappears at a grid point to confer the p-density a numerical value $\sim 10^{-4}$ or $\sim 10^{-5}$. If N_0 clouds are modified, this value is raised to the power N_0 and

$$p(\boldsymbol{\psi}_i^t|\boldsymbol{\psi}_i^{t-1}) \sim \begin{cases} (\lesssim 2.10^{-4})^{N_0} & \text{for a number } N_0 \text{ of changing clouds} \\ \sim 1.0 & \text{for a stationary state of clouds..} \end{cases} \tag{4.39}$$

As to $q(\boldsymbol{\phi}_i^t|\boldsymbol{\psi}_i^{t-1})$, it can take similar values as (4.39). Hence, we have a balance of influences on the weights between the observation density, based on the static distance of the present state vector to the observation, and a p/q-density fraction which follows the clouds' dynamics. This means that the dynamics of clouds can potentially penalize the weight more than the latter would be rewarded by the particle being closer to observations.

Table (4.1) shows such an example where the 'best placed' particle (particle 4) is not endowed with the highest weight, and therefore not reselected at the resampling step. The observation contains two clouds, and particle 4 is the only one among all six particles to have a correct cloud. But its weight, however, is not high compared to others' weights. Although its observation density is the highest among all particles ($\sim 2.10^{-9}$), the fact that a cloud was born in particle 4 is more penalizing than the benefit of having a particle closer to the observation would be rewarding. In Table (4.1), the particles which are resampled are designated by 'resampled particles' (at the bottom of the screen shot). Particle 2 is replicated twice and particle 1 four times. Note that not all equally likely particles are necessarily reselected (for example, 'new particle' 4 is not reselected, although it was not penalized by a change from former to present state). This is because resampling only works well statistically (cf. Chapter 2).

particle	p	q	p/q	obs-density	(norm.) weight
1	0.999769	0.999769	1.0	5.2035e-13	0.32310
2	0.999769	0.999769	1.0	5.2035e-13	0.32310
3	0.999538	0.999538	1.0	9.0281e-16	0.00056
4	2.3097e-5	0.999769	2.310e-5	2.0611e-9	0.02956
5	0.999769	0.9997690	1.0	5.2035e-13	0.32310
6	0.99953	0.9997690	1.0	9.0281e-16	0.00056

(Index of) resampled particles	2	2	1	1	1	1

Table 4.1: Example of an undesired effect of the particle filter on the cloud model with hl 3000 and exponential observation density. The 'ratio' between the observation density and density fraction of evolution of the clouds is detrimental to the particle (particle 4) that is closest to the observation. The reselected and winning particles (particles 1 and 2) are those which did not change since the last time step.

Consequences of Gaussian vs exponential observation density

The nature of the observational density plays an important role in the selection mechanism of the filter as well. As before with $\sigma_e = 0.05$ and $\Delta := \sqrt{(\boldsymbol{\psi}_i^n - \mathbf{d}^n)^2} \neq 0$, and for $\sigma_g = 0.5$, we have either

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) \lesssim 10^{-9\Delta} \quad \text{or} \quad p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) \lesssim 10^{-9\Delta^2}, \tag{4.40}$$

depending whether the observation density is exponential or Gaussian. It is clear that in the competition of the observation density with the transition densities p and q in (4.39), the exponential observation density will have more chance to win in terms of contributing to larger weights, than the Gaussian observation density.

We expect the SIR filter using a Gaussian observation density to lead to smaller RMS errors than using an exponential observation density, since the allowed error on the observations are smaller with a Gaussian observation density. But the efficient particle filter is not expected to work well with a Gaussian observation density, since nudging will potentially penalize weights without the presence of a counterbalance of the observation density to increase them, in view of resampling (cf. Section 4.3).

Consequences of weak and strong nudging

In the cloud model, all states have to be integer valued, and non-negative. Therefore, the states have to be rounded before their densities are computed. Let us use the notations as in (4.24) and consider one grid point. To this purpose, let

$$\{x\} = \begin{cases} \lfloor x + 0.5 \rfloor & \text{if} \quad x \geq -1 \\ 0 & \text{else,} \end{cases} \tag{4.41}$$

where '$\lfloor$' is the smallest and nearest integer to which a real value is rounded. Suppose the **K**-matrix is diagonal. If the particle coincides with the observation (which can happen by chance with or without nudging), we have

$$\psi^n = d^n \Rightarrow \phi^n = \{(1-K)^{-1}(\psi^n - Kd^n)\} = \{\psi^n\} = \psi^n, \tag{4.42}$$

and therefore the p and the q densities are equal no matter which changes in the number of clouds have taken place before, and we have $p = q$. Note that this holds only for one grid point. In this case, the observation density is also equal to 1, and the weights as well. This shows that if the model finds the perfect match to the observation, the past dynamics that led to this match become irrelevant.

Let us rewrite ϕ^n as

$$\begin{aligned} \phi^n &= \{(1-K)^{-1}(\psi^n - Kd^n)\} = \{(1-K)^{-1}(\psi^n - K\psi^n + K\psi^n - Kd^n)\} \\ &= \{\psi^n + \epsilon\Delta\}, \quad \text{where} \quad \Delta := d^n - \psi^n \in \mathbb{Z}, \quad \epsilon := \frac{K}{1-K} > 0. \end{aligned} \tag{4.43}$$

If $\Delta := \psi^n - d^n \neq 0$, then due to

$$\epsilon|\Delta| > \frac{1}{2} \Leftrightarrow K > \frac{1}{1+2|\Delta|}, \tag{4.44}$$

only a nudging parameter of $K > \frac{1}{1+2|\Delta|}$, where $|\Delta| = 1, 2, ...$, will enforce a change of $\phi^n = \psi^n$ into $\phi^n \neq \psi^n$ in (4.43). In our present model, $|\Delta|$ will mostly take the values 1, or very rarely 2. Then,

$$\phi^n \neq \psi^n \Leftrightarrow K > \frac{1}{3} \tag{4.45}$$

Such a difference between ϕ^n and ψ^n that is induced by nudging leads *either* to a *penalization* or to a *rewarding*, i.e. to a decrease or to an increase of the weights. Indeed, if $\psi^{n-1} = \psi^n$ and $\psi^{n-1} \neq \phi^n$, it follows according to (4.32) that $p = 1$ and $q \leq 10^{-4}$, so that $p/q \geq 10^4$ considerably increases the weight, whereas if $\psi^{n-1} \neq \psi^n$ and $\psi^{n-1} = \phi^n$, then $q = 1$ and $p \leq 10^{-4}$ and thus $p/q \leq 10^{-4}$ decreases it. This is how the fraction p/q, by the effect of the dynamics and of nudging, interacts with the static observation density, which itself only depends on the present distance between state vector and observation. On the other hand, nudging too much ($K \to 1$) will lead to $\epsilon \to \infty$ and hence inforce too large a change between ϕ^n and ψ^n, which is totally unlikely and will make the filter collapse onto one single particle.

It follows that for weak nudging, the density fraction p/q is most often equal to 1 and that the filter is governed by the observation density, and hence by the static position of the particles with respect to the observation like in the SIR filter. For stronger nudging, we get a fraction p/q different from 1 which depends on the dynamics and on nudging of the model. An enforced change of the model through nudging then enters the weight's value and competes with the observation density, possibly altering the particle's weight, penalizing or increasing them (cf. Table (4.2)).

Table (4.2) shows an example of how it comes that a worse placed particle wins in terms of weights over a particle that is closest to observations. Although particle 4 has nearest distance to the observation with an observation density of 10^{-13} (compared to particle 3 with 'obs-dens' $\sim 10^{-18}$), the dynamics induced by the medium nudging $K = 0.5$ in particle 3 through a p/q-fraction of $\sim 2.5 \times 10^6$ leads to an increase of the latter's weight. Thus, particle 3 is endowed with the highest weight and is the only particle that is replicated ten times (cf. the bottom of Table (4.2), where 'resampled particles' the list of indices of particles which are being reselected in the resampling procedure).

particle	p	q	p/q	obs-density	(norm.) weight
1	0.00067	1.69e-7	3951.82	1.045e-23	4.13e-20
2	4.65e-8	1.70e-10	272.52	5.29e-22	1.44e-19
3	2.58e-7	1.01e-13	2547715.84	4.24e-18	1.08e-11
4	0.00271	0.00663	0.40	5.20e-13	2.12e-13
5	4.92e-6	4.85e-8	101.37	2.70e-25	2.74e-23
6	2.38e-6	1.05e-7	22.48	1.04e-23	2.35e-22
7	6.01e-7	9.79e-11	6142.73	8.15e-31	5.00e-27
8	3.27e-5	05.53e-5	0.59	9.0281e-16	5.34e-16
9	4.00e-9	5.93e-11	67.46	2.70e-25	1.82e-23
10	2.39e-7	7.98e-8	2.99	1.53e-36	4.60e-36

(Index of) resampled particles	3	3	3	3	3	3	3	3	3	3

Table 4.2: Example of the effect of the dynamics of clouds for a stationary cloud field which is subject to medium nudging $K = 0.5$. The best placed particle (particle 4) is not endowed with the highest weight. It looses the competition against a worse placed particle (particle 3) whose dynamics is more beneficial to the final weight.

This example shows the influence of nudging on the weights through the p/q density fraction, which can be beneficial as well as detrimental to the weights. But we expect such disadvantages encountered by better placed particles to be counterbalanced by the convergence of the filter towards the observations, provided that this convergence has time to last within a stationary regime of observations (see Results of this Section). For if we nudge enough, the potential gratification of the weight induced by nudging will lead to the enhanced selection of particles that are nudged and thus, better placed.

Dichotomic behaviour of the weights

Based on the above discussion, we expect a dichotomic situation to take place during the computation of the weights. No matter in which regime of fast or slow evolving clouds we are, we have always one of the following situations. Either

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) = 1.0 \quad \text{and} \quad p/q = 1.0, \tag{4.46}$$

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) = 1.0 \quad \text{and} \quad p/q \neq 1.0, \tag{4.47}$$

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) \neq 1.0 \quad \text{and} \quad p/q = 1.0, \tag{4.48}$$

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^n) \neq 1.0 \quad \text{and} \quad p/q \neq 1.0. \tag{4.49}$$

The first case, where the model hits the observation by chance and without nudging or with weak nudging, nearly never happens. Or it is already a sign of filter collapse, when the particles are forced onto the observations. The second case happens mainly for strong nudging. The model hits the observations after nudging which is strong enough to modify the weights by penalizing or increasing them[2]. The third is encountered for weak nudging. The model does not approach the observation and nudging is too weak to add a contribution to the weights, since ϕ is not made different from ψ by nudging such as to make the p/q-fraction different from 1. The filter is basically ruled by the SIR filter. The fourth case corresponds to medium nudging, where the efficient filter is actually manifesting its 'contribution'. Neither the observations are hit, nor are the weights determined by the static observation density only. But these cases should be considered statistically, valid for each of the grid points and each of the particles. Only a large number of cases, summing up all the behaviours encountered according to these 'rules', leads to statistically relevant result of the particle filter.

The weights are thus governed by a dichotomy, within each of the two density types (observation density and the p/q-fraction), that consists of being equal to one, or not. The interaction between both types of densities depends on nudging and holds for any regime of cloud variability, or initial cloud density. And although the cloud variablity, i.e. the velocity with which the cloud numbers vary in time (via the cloud half life) does not appear to have a directly visible effect on the densities and the weights, we expect it to indirectly influence in a very strong way the filter behaviour. Indeed, a *slow varying cloud field seems necessary* for the filter with medium nudging to converge to the observations *and remain close to them* without collapsing onto them, so that *only better placed particles* are given a higher weight. This is only possible if they were made closer to the observation by (strong enough) nudging, and under these conditions it makes an improvement with respect to the simple SIR filter very likely.

4.2 Test settings

The test settings are based on specifications in [59]. We reproduce the SIR filter results and compare them to the efficient particle filter with nudging by computing the root mean square (RMS) error.

Parameters

The number of grid points is $N = 100$. The number of integration time steps is $nsteps = 100$. The mean density of clouds per grid point is $\rho = 0.1$ This means that there are ten clouds on 100 grid points on average. The cloud half lives are fixed to either $hl = 30$, $hl = 3000$ or $hl = 30000$ time steps, representing slow varying and faster varying clouds.

[2] The weights are also penalized by a p/q-fraction which is much larger than one, because then all other particles are discarded to the favour of this selected particle, which might not at all be closer to the observations. The filter then collapses on a wrong particle.

Table 4.3: Parameter setting for the stochastic cloud model

parameter		
N	100	number of grid points
k	50	number of particles
t	100	number of assimilation time steps
σ_e	0.05	parameter of the exponential observation distribution
ρ_0	0.1	initial cloud density, parameter of the Poisson process
hl	30000	cloud half life 30000
λ	$\approx 2.10e^{-6}$	birth probability of a cloud
μ	$\approx 2.10^{-5}$	death probability of any existing cloud
hl	3000	cloud half life 3000
λ	$\approx 2.10e^{-5}$	birth probability of a cloud
μ	$\approx 2.10^{-4}$	death probability of any existing cloud
hl	30	cloud half life 30
λ	$\approx 2.10e^{-3}$	birth probability of a cloud
μ	$\approx 2.10^{-2}$	death probability of any existing cloud

Model integration

One realization of the model is integrated as truth for 100 time steps. First, we draw an initial number of clouds at each grid point following the Poisson distribution. For later integration timesteps, we generate a new number of clouds at each grid point according to the birth-death process (cf. 3.1.1), whereby either a cloud is added at each grid point with probability λ, or each of already existing clouds disappears with probability μ (cf. (3.3)). Another set of k realizations of the model is integrated to constitute the ensemble of particles, where one particle is defined as N-dimensional vector containing the different numbers of clouds.

For the assimilation experiment, a set of $k = 50$ particles is integrated over time. An additional realization is generated as truth. The particles are approached towards the observations by an arbitrary forcing via the nudging parameter K, where K is the constant diagonal element of the nudging matrix $\mathbf{K}$ in equation (4.21). This parameter logically takes values lying between 0 and 1, since $K > 1$ would lead to an 'overshooting' of the particles beyond the observations, departing them again from the latter. The birth-death process by itself and the fractional value of K are a source of non-integer numbers in the forward integration of the model, i.e. the number of clouds at each grid point. Therefore this number is set to integer within the model integration by rounding up or down the fractional value of the model to the next higher or lower integer. This is how, even when we use fractional K-values, the model is always integer valued.

Observations

The observations are chosen for simplicity to be identical to the truth run. Observations are available at each time step and at every grid point. Assimilation tests were made for two different observation densities, namely for Gaussian and for Exponential observation error distributions. Let us denote for the following the subscript g for the Gaussian distrubution and e for the Exponential distribution. The Exponential observation distribution in (4.28) was used following [59] as

$$p_2 := p(\mathbf{d}^n|\boldsymbol{\psi}_i^t)_{\exp} = C_1 \exp\left(-\frac{\sqrt{(\boldsymbol{\psi}_i^t - \mathbf{d}^n)^2}}{\sigma_{\mathrm{e}}}\right), \tag{4.50}$$

where $\boldsymbol{\psi}_{tr}^t$ is the truth-vector at time t, $i \in [1,k]$ is the particle index and where C_1 is a constant. $\sigma_{\mathrm{e}} = 0.05$ is the parameter of the exponential distribution. The Gaussian observation distribution is given by

$$p_1 := p(\mathbf{d}^n|\boldsymbol{\psi}_i^t)_{\mathrm{gauss}} = C_2 \exp\left(-\frac{1}{2}\frac{(\boldsymbol{\psi}_i^t - \mathbf{d}^n)^2}{\sigma_{\mathrm{g}}^2}\right), \tag{4.51}$$

where σ_{g}^2 is the observation error variance, and C_2 a constant. The constants C_1 and C_2 can be neglected in the computation of the weights because they are a common factor of the weights to all particles, and they do not change the latter since these weights get normalized at each resampling step.

In order to find the variance σ_{g}^2 corresponding to the parameter σ_{e} in the Exponential distribution, however, we need to compute the constants C_1 and C_2 so that separately we have $C_1 \int p_1 d\boldsymbol{\psi} = 1$, $C_2 \int p_2 d\boldsymbol{\psi} = 1$. Then, we will equalize

$$C_1 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 p_1(\boldsymbol{\psi}) d\boldsymbol{\psi} = C_2 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 p_2(\boldsymbol{\psi}) d\boldsymbol{\psi}. \tag{4.52}$$

From this requirement we will derive the relationship between σ_{e}^2 and σ_{g}^2 [3]. Let us compute the constants C_1 and C_2.

Using a change of variables into N-dimensional spherical coordinates, we have

$$1 = C_1 \int_{\mathbb{R}^N} e^{-\frac{\boldsymbol{\psi}^2}{2\sigma_g^2}} d\boldsymbol{\psi} = C_1 \int_{\mathbb{R}^+} e^{-\frac{r^2}{2\sigma_g^2}} r^{N-1} S^{N-1} dr, \tag{4.53}$$

where $S^{N-1} = \frac{2\pi^{\frac{N}{2}}}{\Gamma(\frac{N}{2})}$ is the surface of an N-dimensional sphere of radius r. The consecutive

[3] The choice of equating the mean square of these variables in order to find the relationship between the Gaussian variance and the exponenial density parameter comes from the fact that equating only the variables' mean would not answer the question since the Gaussian vector's mean is zero, $E[\boldsymbol{\psi}]_{\mathrm{gauss}} = \mathbf{0}$. Neither could the normalized integrals be equalized, since they are both equal to one.

change of variables $\frac{r^2}{2\sigma^2} = y$ leads to

$$\begin{aligned} 1 &= C_1 S^{N-1} 2^{\frac{N}{2}-1} \sigma_g^N \int_{\mathbb{R}^N} y^{\frac{N}{2}-1} e^{-y} dy \\ &= C_1 S^{N-1} 2^{\frac{N}{2}-1} \sigma_g^N \Gamma(\frac{N}{2}), \end{aligned}$$

and hence to

$$C_1 = \frac{1}{S^{N-1} 2^{\frac{N}{2}-1} \sigma_g^N \Gamma(\frac{N}{2})}. \tag{4.54}$$

In the same way, we compute C_2. We have

$$\begin{aligned} 1 &= C_2 \int_{\mathbb{R}^N} e^{-\frac{|\boldsymbol{\psi}|}{\sigma_e}} d\boldsymbol{\psi} \\ &= C_2 \int_{\mathbb{R}^+} e^{-\frac{r}{\sigma_e}} r^{N-1} S^{N-1} dr \\ &= C_2 S^{N-1} \sigma_e^N \int_{\mathbb{R}^+} y^{N-1} e^{-y} dy \\ &= C_2 S^{N-1} \sigma_e^N \Gamma(N) \end{aligned}$$

after the change of variables $\frac{r}{\sigma_e} = y$. Consequently,

$$C_2 = \frac{1}{S^{N-1} \sigma_e^N \Gamma(N)}. \tag{4.55}$$

Now we can derive the relationship between σ_e and σ_g from (4.52). Using the same changes of variables as before, we have

$$\begin{aligned} C_1 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 p_1(\boldsymbol{\psi}) d\boldsymbol{\psi} &= C_2 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 p_2(\boldsymbol{\psi}) d\boldsymbol{\psi} \\ \Leftrightarrow C_1 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 e^{-\frac{\boldsymbol{\psi}^2}{2\sigma_g^2}} d\boldsymbol{\psi} &= C_2 \int_{\mathbb{R}^N} \boldsymbol{\psi}^2 e^{-\frac{|\boldsymbol{\psi}|}{\sigma_e}} d\boldsymbol{\psi} \\ \Leftrightarrow C_1 \int_{\mathbb{R}^N} r^2 e^{-\frac{r^2}{2\sigma_g^2}} r^{N-1} S^{N-1} dr &= C_2 \int_{\mathbb{R}^N} r^2 e^{-\frac{r}{\sigma_e}} r^{N-1} S^{N-1} dr \\ \Leftrightarrow C_1 S^{N-1} 2^{\frac{N}{2}} \sigma_g^{N+2} \int_{\mathbb{R}^+} y^{\frac{N}{2}} e^{-y} dy &= C_2 S^{N-1} \sigma_e^{N+2} \int_{\mathbb{R}^+} y^{N+1} e^{-y} dy \\ \Leftrightarrow C_1 2^{\frac{N}{2}} \sigma_g^{N+2} \Gamma(\frac{N}{2}+1) &= C_2 \sigma_e^{N+2} \Gamma(N+2). \end{aligned}$$

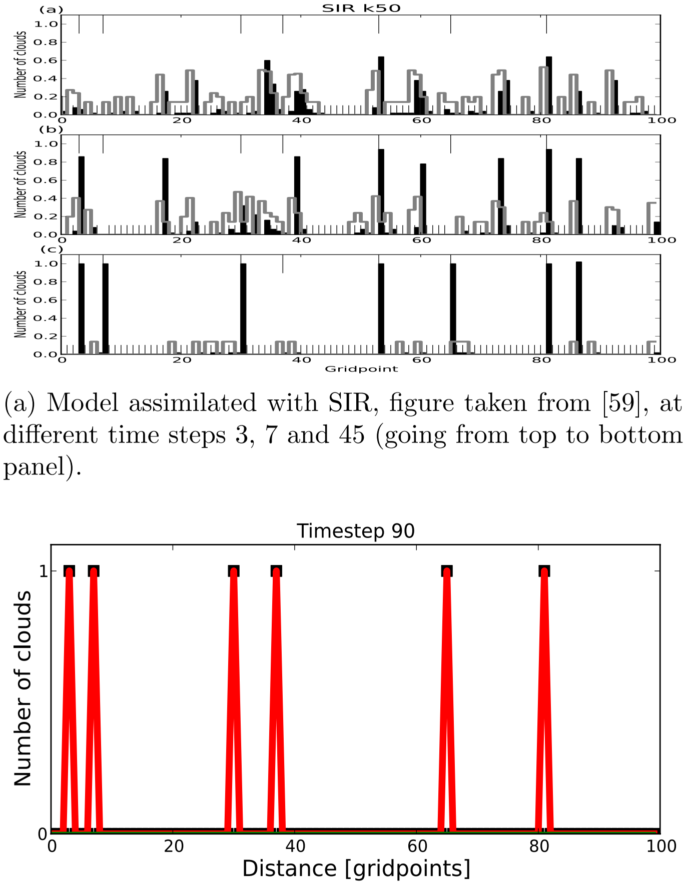
After substituting the two constants (4.54),(4.55) and making use of the expression $\Gamma(x+1) = \Gamma(x)x, x \in \mathbb{Q}$, the former equality finally simplifies to

$$\sigma_g = \sigma_e \sqrt{N+1}, \tag{4.56}$$

where N is the size of the domain. Hence, if we have $N = 100$ grid points, the Gaussian standard deviation is
$\sigma_g = 0.05 \times \sqrt{101} \approx 0.5$.

4.3 Results

Based on the experiments of Würsch and Craig in [59], we apply the SIR filter to the stochastic cloud model and compare it to the efficient particle filter with different nudging values. Unlike for the shallow water model (see next chapter), this filter uses no weight equalization procedure as in [33]. Two types of tests are made to apply the filter. The first one consists of comparing the filter behaviour for fast and slowly varying clouds, the second one compares a Gaussian with an exponential observation error distribution, as specified in the previous section (Sec. 4.2).



(a) Model assimilated with SIR, figure taken from [59], at different time steps 3, 7 and 45 (going from top to bottom panel).

(b) Model assimilated with the efficient particle filter at time step 90

Figure 4.1: Comparison between the cloud model assimilated with the SIR filter (a) and the efficient particle filter (b). The bottom panel in (a) shows the model at time step 45 using the SIR filter. Not all observed clouds (thin vertical lines on the upper part of the panels in (a)) are captured by the SIR-filter, and erroneous clouds are produced that do not correspond to observations. By comparison, the efficient particle filter (b) using intermediate nudging $K = 0.5$ (and an exponential observation density) captures all observations, without spurious cloud production. The observations in (b) are represented by black dots.

Figure 4.1 a) (taken from [59]) shows the result of the SIR filter at three consecutive time steps. The SIR filter manages to capture some clouds but not all, and produces erroneous clouds where no clouds are observed. The spread of the ensemble is right away reduced, the filter replicating very few particles. In contrast, the filter with a nudging forcing of $K = 0.5$ (Figure 4.1b) illustrates the rapid convergence of the filter towards the observed clouds. But for this strong nudging value, the filter collapses to only one or at most two particles.

In the following experiments, different values for the nudging parameter K (cf. the legends of the figures below) are tested to analyze the response of the filter. We show figures of the mean RMS error and the mean spread, through time, of the particles after applying the efficient particle filter without weight equalization. The RMS error at assimilation time t was computed as

$$\mathrm{RMS}(t) = \sqrt{\overline{(\boldsymbol{\psi}_i(t) - \boldsymbol{\psi}_{\mathrm{tr}}(t))^2}}, \quad i \in [1, k], \tag{4.57}$$

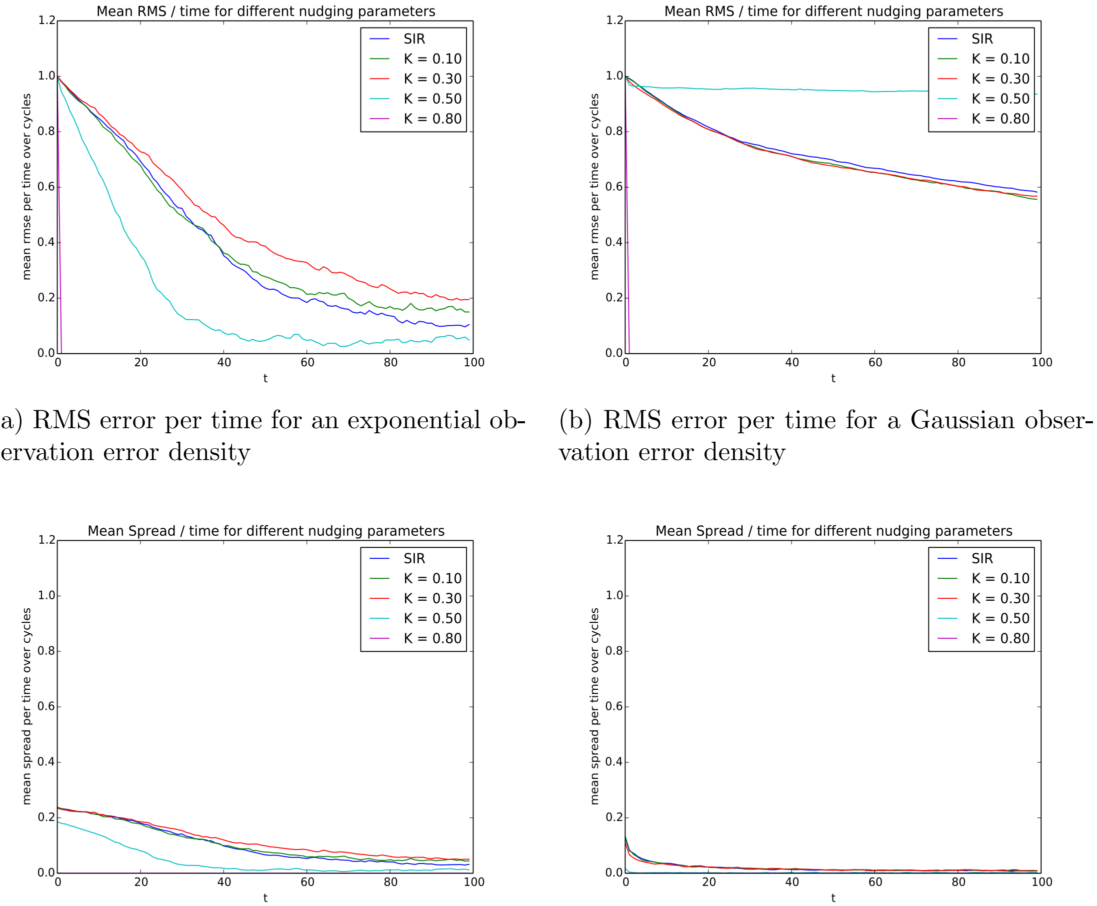
where $\boldsymbol{\psi}_{\mathrm{tr}}$ is the truth and where the mean is taken over the ensemble of k members. The spread (or standard deviation) of the particles is defined as

$$\sigma(t) = \sqrt{\overline{(\boldsymbol{\psi}_i(t) - \overline{\boldsymbol{\psi}_i}(t))^2}}, \quad i \in [1, k]. \tag{4.58}$$

Both the RMS error and the spread were averaged over 300 repetitions of the experiment, and both normalized by the value of the RMS error of the very first time step if the latter error was non-zero, replaced by 1 otherwise.

4.3.1 Cloud variability and observation error density

Figure 4.2 shows the mean RMS error (panels a and b in Figure 4.2) and the mean spread (panels c and d) per time for a very slowly varying cloud field with half life $hl = 30000$. A cloud has a 'life expectation' of 30000 integration time steps. The panels on the left represent an exponential observation error density, while the right hand side panels were made with use of a Gaussian observation error density. For the exponential observation error density (Figures 4.2 a and c), the SIR-filter performs only slightly better than the efficient particle filter after 40 time steps, for a weak nudging parameter of $K = 0.1$, while nudging moderately ($K = 0.3$) leads to a larger error. This can be explained as a consequence of the considerations in Section (4.1.2), namely that integer valued model states represent an issue for nudging. Nudging less than one half will be rounded down to not nudging the state at all, at the same time leading to a shift of ϕ that makes it different from ψ and penalizes, through $p/q \ll 1$, the corresponding weight without even having the benefit of bringing the states closer to the observation.



(a) RMS error per time for an exponential observation error density

(b) RMS error per time for a Gaussian observation error density

(c) Spread per time for an exponential observation error density

(d) Spread per time for a Gaussian observation error density

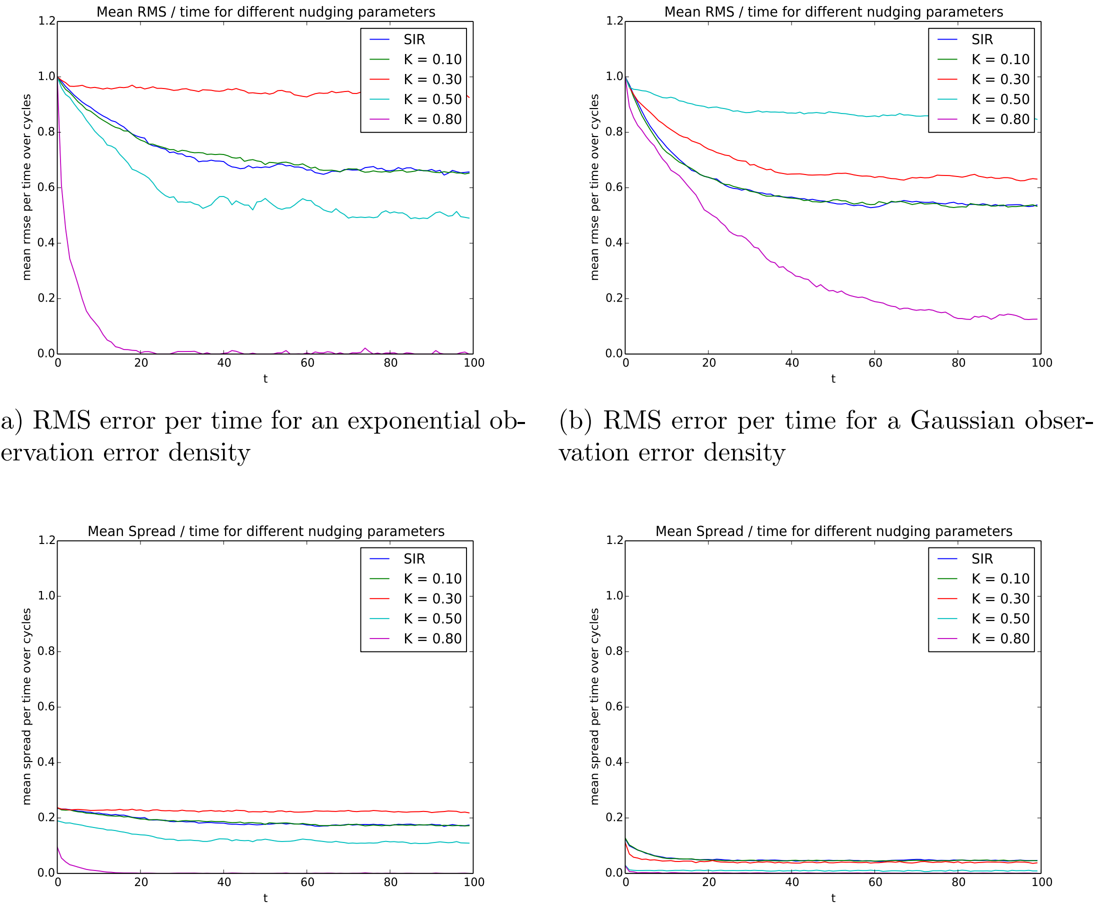
Figure 4.2: RMS error and spread per time for an exponential observation error distribution (panels a and c), vs a Gaussian error distribution (b and d), and cloud half life $hl = 30000$.

Only nudging beyond a threshold of $K \sim 0.3$ leads to an improvement of the efficient particle filter with respect to SIR, but at the cost of a fast decreasing and very reduced spread and thus of diversity of the ensemble. Nudging close to 1.0 is prohibited because of the filter collapse (here $K = 0.8$).

The filter with the Gaussian observation density shows a much slower convergence than the one for the exponential density, for all nudging values but a too strong one of $K = 0.8$. Nudging weakly leads to a similar behaviour of the efficient filter compared to the SIR filter, whereas medium nudging ($K = 0.5$) drastically increases the error, before filter collapse happens for strong nudging ($K = 0.8$). The binary behaviour of nudging first without effect or with sudden penalization, expressed through the weights and a larger RMS error, and consecutive filter collapse after too strong relaxation, is visible in this example of a

Gaussian observation distribution as well. The spread is significantly reduced compared to the filter with an exponential observation density. It shows that the efficient particle filter fails for strong nudging and does not surpass the SIR filter, for this discrete model and Gaussian observation errors.

Figure 4.3 shows the mean RMS error and spread per time for a slightly faster evolving cloud field with half life $hl = 3000$, for an exponential observation error density (Figures (4.3 a) and c) vs a Gaussian one (Figures 4.3 b) and d). The same features as before can be observed here in the exponential case. Nudging weakly will either lead to an equivalent result as SIR ($K = 0.1$), or to a worse result than no nudging, while after a threshold of $K \sim 0.5$, nudging becomes beneficial and more efficient than SIR.



(a) RMS error per time for an exponential observation error density

(b) RMS error per time for a Gaussian observation error density

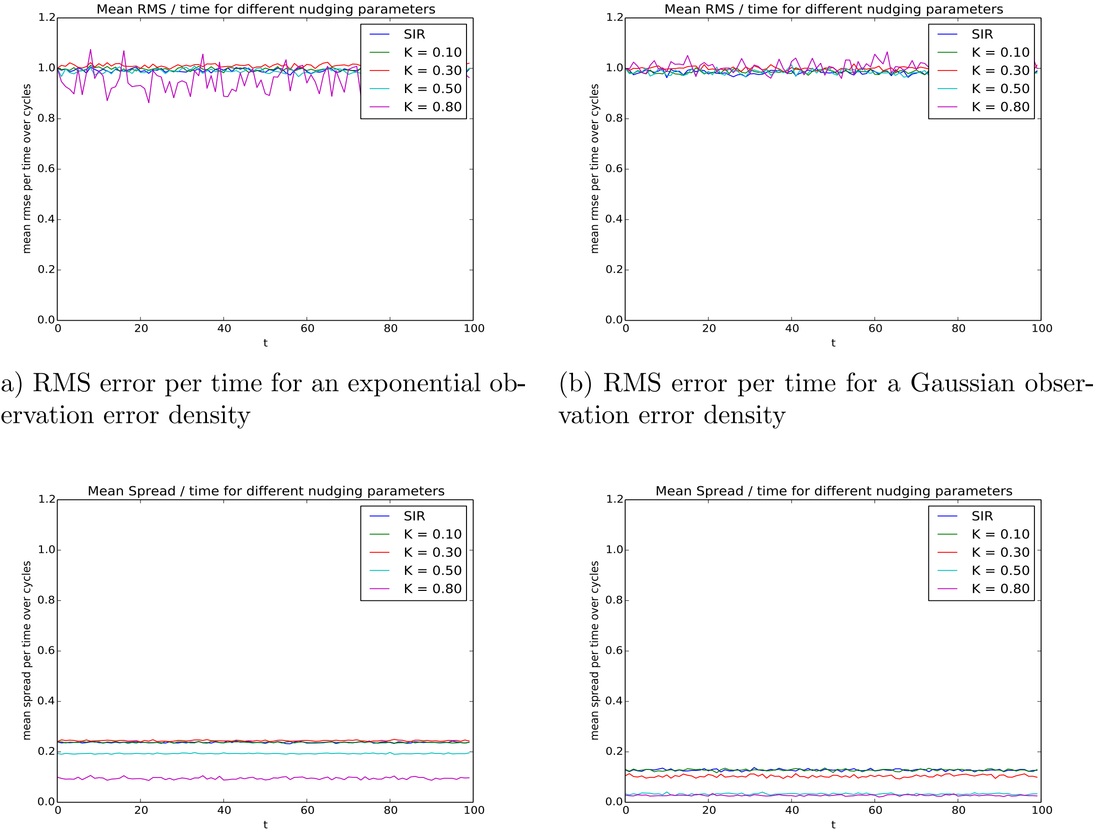
(c) Spread per time for an exponential observation error density

(d) Spread per time for a Gaussian observation error density

Figure 4.3: RMS error and spread per time for an exponential observation error distribution (left panels) vs a Gaussian one (right panels) for cloud half life $hl = 3000$

It takes all variants of the filter (SIR and efficient filter with nudging, apart from nudging strongly with $K = 0.8$) fewer time steps to reach a stationary error level as in Figure 4.2, but the smallest possible error reached after 100 time steps of assimilation is much higher than before. The best result is obtained with an intermediate nudging ($K = 0.5$), which surpasses the SIR filter by nearly 25 percent after 100 time steps. The spread is still low, but higher than in the former, stationary case. More nudging goes hand in hand with reducing diversity of particles, unless a weight equalization procedure as in [33] is used.

In the Gaussian case (Figure 4.3 b) and d) , there is again a very clear dichotomy of either doing worse than SIR with weak to medium nudging, or better than SIR at the cost of filter collapse. The efficient particle filter is not able to outperform SIR. The weights are penalized by nudging without being able to counterbalance this negative effect by the values of the observation density.



(a) RMS error per time for an exponential observation error density

(b) RMS error per time for a Gaussian observation error density

(c) Spread per time for an exponential observation error density

(d) Spread per time for a Gaussian observation error density

Figure 4.4: RMS error and spread per time for an exponential observation error distribution (panels a and c), vs a Gaussian error distribution (b and d), and cloud half life $hl = 30$.

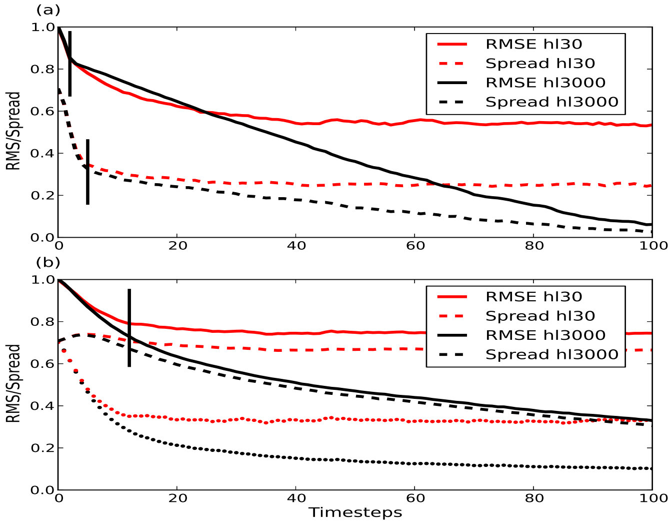
We infer this behaviour to the fact that the Gaussian exponents of the observation density are much smaller than the exponents of the exponential observation density. Hence, the observations are more easily penalized in the p/q-density fraction by nudging than they were with higher observation density contributions which were able to cover small negative effects of nudging.

A fast varying cloud field (hl 30, Figure 4.4) does not bring any filter convergence anymore. The filter stays at a constant level of error in time for any nudging but a strong one, which leads to filter divergence. The high stochastic variability ensures a larger spread than in the examples shown above. The filter is not able to follow the observations in time.

Figure 4.5 (taken from [59]) shows the former results for a SIR experiment made in [59] for an observation error density which gives the observations more weight than the exponential density, since it is given by

$$p(\mathbf{d}^n|\boldsymbol{\psi}_i^t) \sim e^{-\frac{\sqrt{(\mathbf{d}^n-\boldsymbol{\psi}_i^t)^2}}{\sigma_e\sqrt{N}}}, \tag{4.59}$$

where $\sigma_e = 0.05$ and N is the number of grid points. Compared to the densities used in the previous experiments above, this type of density has the largest exponent. In this way, the observation density has more impact on the weights than the p/q density fraction. The RMS error reaches a smaller value than using an exponential observation density (cf. Fig. 4.3) after 100 time steps for $hl = 3000$. A comparable result can be achieved by the exponential density for a cloud half life which is ten times larger than 3000 (cf. Fig. 4.2).



(a) Taken from [59]

Figure 4.5: RMS error and spread per time for a) SIR and b) ETKF

4.3.2 Conclusion

The Stochastic Cloud model developed by Würsch and Craig [59] was a first simplified model within a hierarchy of test models for convective scale data assimilation. It contains minimal requirements for the simulation of a convective environment by revealing highly nonlinear and stochastic features. It has non-Gaussian error statistics, and since it is purely stochastic, it is an extreme testbed for nonlinear data assimilation. Its main non-realistic features are the lack of correlations in the errors (clouds appear and die at each grid point independently from each other), the missing deterministic physical dynamics and its integer-valued model states.

We showed that the efficient particle filter with nudging was very sensitive to the observation density, which plays an important role in the selection mechanism of the weights. The larger the negative exponents in the observation density are, the smaller is the impact of the position of particles on the selection mechanism, and the less chance the efficient filter has to work at all, without causing filter collapse. This was demonstrated using a Gaussian observation density. The efficient particle filter also appears to be very sensitive to the cloud half life, i.e. to the variability of clouds in time. Indeed, fast varying cloud fields do not leave the filter any possibility to follow the observations, suggesting that the filter needs time to adapt to the dynamics of the observations. On the other hand, we could show that the cloud life time directly enters the computation of the probability densities p and q and that it regulates the frequency with which the latter compete with the observation density. The faster the clouds change, the more the p and q densities will contribute to perturb the influence of the static distance between state and observation which *de facto*, decides upon the RMS error.

The requirement of the model to take only integer values makes it difficult to apply nudging, the main ingredient of the efficient particle filter, which was designed to be gradual in intensity. It is impossible to nudge an integer valued particle 'just a little' or 'a little more'. It is either 'all' or 'nothing'. Below a certain threshold, no particle is nudged and above it, all particles are just replaced by the observation. This dichotomy imposed by rounding also reveals itself in the corresponding Importance density. It is the probability density of a rounded state that is evaluated. Although this density is specially adapted to integer values, a whole range of possible values that gradual nudging would allow is thus cast off from consideration. The constraints that the stochastic cloud model imposes on the filter exhibit the former's limits with respect to real conclusions that could be made on the behaviour of both model and filter for a realistic 'real-world' application.

However, this simple toy model helped to gain significant insight into the problems, applicability and functioning of the particle filter that may arise if tested in the future. By deriving the exact probability density according to the clouds' generation, which usually is impossible to do for NWP-models, we could illustrate how a non-Gaussian model density impacts the weights and how it interacts with a given Gaussian or exponential observation error density. In this context, we showed that there is a dichotomy of influences entering the weights' computation, which consists of a static probability density represented by the observation error density, and a time evolution probability density that takes into account the dynamics of the model. Both the static and the dynamical densities compete with each other unless one of both is 'swiched off', either by nudging too much (hence leading to a collapse of the filter onto the observations) or not enough or by not nudging at all (making the weights only depend on the observation density, i.e. on the static location of the last time step model states, as in the SIR filter).

Chapter 5

Testing the modified Shallow water model

In this chapter, we test the efficient particle filter of Van Leeuwen [33] on the modified shallow water model [59] introduced in Chapter 3.1.2. The shallow water model is the second in the hierarchy of testbed models for convective scale data assimilation, after the stochastic cloud model (cf. Sections 3.1.1, 4). In addition to the features of nonlinearity and non-Gaussianity that characterize convection, it represents a physically more realistic model than the stochastic cloud model since it contains correlations between grid points and a deterministic evolution part given by partial differential equations. While in the stochastic cloud model, the only observed variable was the number of clouds, we have in the shallow water model three interacting atmospheric variables, wind velocity, water height and rain, which are involved through a set of fluid dynamical equations.

The modified shallow water model has continuous values, and nudging can be performed gradually. Hence, applying the efficient particle filter on this model is likely to give more insight into the effects of nudging, on the weights and on the model behaviour, compared to the stochastic cloud model. The error of the shallow water model is not determined by a purely stochastic evolution of the model anymore, but by a sum of a physically deterministic and a stochastic component. The perturbation is chosen to be Gaussian at each integration time step. Therefore, we expect the efficient particle filter to work better than the SIR filter, compared to the stochastic cloud model. An important unknown, though, is the maximum nudging that the model allows, before too strongly 'pushed' model variables create imbalances in the model, rendering nudging more harmful than beneficial.

In the following sections, we consider two main questions:
1.) What is the effect of a spatial reduction in the number of observations and in the dynamical variables observed?
2.) How does the choice of the nudging matrix (diagonal or non-diagonal) affect the Root Mean Square (RMS) error of the model with respect to the (synthetic) truth?

Section 5.1 describes how the design of the stochastic perturbation and some parameters in the original shallow water model [59] had to be adapted so that the dynamical variables could be assimilated with the efficient particle filter. The parameters of the model and of the assimilation settings are defined in Section 5.2.

Section 5.3 presents the results of the efficient particle filter tested on the shallow water model. While Section 5.3.1 gives a first glance of results observed in the easiest and most unrealistic frame of test settings, Sections 5.3.2 and 5.3.3 answer the questions posed above. They show that the number of assimilated observations or variables, as well as the shape of the nudging matrix, have an impact on the results. Under conditions with less information - as they are encountered in real situations in NWP - the efficient particle filter leads to smaller *relative* errors compared to a nudged ensemble without resampling, or to SIR.

In Section 5.4, we study the selection mechanism of the resampling procedure that is used in the filter. We show that this mechanism has a chance to outperform a regime of 'pure nudging' only under certain conditions involving the exponent of the particles' weights. We derive some analytical conditions for parameters of the method for which the resampling method is expected to perform well.

5.1 Adapting the stochastic perturbation design to the filter

In the original model conceived by Würsch and Craig in [59], a small noise was added only in the wind equation (cf. β_u in equation (3.9)). This small background noise in u generates enough interaction between the variables velocity (u), water height (h) and rain (r) to produce significantly developed 'cloud' features in terms of rain and water height increase. Ref. [59] introduced an 'arbitrary' perturbation of the wind field which resembles a sinus-curve distributed over four grid points, in order to simulate a small spatial correlation. In their set up, only one such perturbation over $N = 1000$ grid points was added to the wind field at every integration time step.

Resampling in the filtering procedure takes only into account the information of the model that is given by its stochastic part. Hence, our desire of including as much physical information of the model in the weights as possible led us to add noise in all model variables for our purpose of testing the particle filter with nudging. Equation (3.9) becomes then

$$\begin{aligned}
&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial(\phi + \gamma r)}{\partial x} - K\frac{\partial^2 u}{\partial x^2} = \beta_u, \\
&\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} - K\frac{\partial^2 h}{\partial x^2} = \beta_h \\
&\frac{\partial r}{\partial t} + u\frac{\partial r}{\partial x} - K_r\frac{\partial^2 r}{\partial x^2} + \alpha r + \beta\frac{\partial u}{\partial x} = \beta_r,
\end{aligned} \tag{5.1}$$

where $\beta_u, \beta_h, \beta_r$ on the r.h.s are the terms of stochastic forcing of the model and where the same conditional cases for convection initiation as before hold. To suppose noise in all variabe trajectories is also usual in data assimilation, since the truth of any physical variable is not fully known.

Leaning on Van Leeuwen's [33] choice of model noise, we add a Gaussian perturbation to all three variables u, h and r in (5.1) at each integration time step of the model, namely β_u, β_h and β_r. In the numerical implementation, the forcing vectors are

$$\boldsymbol{\beta}_u \sim \mathcal{N}(0, \mathbf{Q}_u), \quad \boldsymbol{\beta}_h \sim \mathcal{N}(0, \mathbf{Q}_h), \quad \boldsymbol{\beta}_r \sim \mathcal{N}(0, \mathbf{Q}_r) \tag{5.2}$$

where for $j \in \{u, h, r\}$, the covariance matrix with four off-diagonals of the form

$$\mathbf{Q}_j = q_j \begin{pmatrix}
1 & \frac{1}{2} & \frac{1}{4} & 0 & \dots & \dots & 0 \\
\frac{1}{2} & 1 & \frac{1}{2} & \ddots & 0 & \dots & 0 \\
\frac{1}{4} & \frac{1}{2} & 1 & \ddots & & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \frac{1}{4} \\
\vdots & \ddots & \ddots & \ddots & \ddots & 1 & \frac{1}{2} \\
0 & \dots & \dots & 0 & \frac{1}{4} & \frac{1}{2} & 1
\end{pmatrix} \in \mathbb{R}^{N \times N} \tag{5.3}$$

generates a noise with correlations between variables over four grid points. The correlation magnitude is designed here to diminish exponentially with increasing distance between grid points. The model error covariance matrix can be written in the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_u & 0 & 0 \\ 0 & \mathbf{Q}_h & 0 \\ 0 & 0 & \mathbf{Q}_r \end{bmatrix} \in \mathbb{R}^{3N \times 3N}. \tag{5.4}$$

5.2 Test settings and parameters

Truth and ensemble model runs

Our data assimilation tests are based on the numerical implementation of the modified shallow water model [59]. The initial conditions for the truth trajectories of $\mathbf{u}$, $\mathbf{h}$ and $\mathbf{r}$, as well as for the ensemble members ($(\mathbf{u}, \mathbf{h}, \mathbf{r}) \in \mathbb{R}^{3N}$) are

$$\begin{aligned} \forall l \in [1, N], \quad u(x_l, 0) &= u_0 = 0.1 \\ h(x_l, 0) &= h_0 = 90.0 \\ r(x_l, 0) &= r_0 = 0.0. \end{aligned} \tag{5.5}$$

The boundary conditions are periodic, as mentioned before.

The truth run is generated by adding a Gaussian perturbation vector $\boldsymbol{\beta}^j$ to the model variable $\mathbf{x}_{\text{tr}} := (\mathbf{u}, \mathbf{h}, \mathbf{r}) \in \mathbb{R}^{3N}$ at each integration time step j, according to the original model equation (3.17),

$$\mathbf{x}_{\text{tr}}^j = f(\mathbf{x}_{\text{tr}}^{j-1}) + \boldsymbol{\beta}^j, \quad \text{where} \quad \boldsymbol{\beta}^j \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \in \mathbb{R}^{3N}. \tag{5.6}$$

This perturbation vector at every small integration time step is chosen according to (5.3), where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_u & 0 & 0 \\ 0 & \mathbf{Q}_h & 0 \\ 0 & 0 & \mathbf{Q}_r \end{bmatrix} \in \mathbb{R}^{3N \times 3N} \tag{5.7}$$

is the error covariance matrix of the original model. With the notations of (3.17), the main diagonal elements of these covariance sub-matrices are

$$\begin{aligned} q_u &= 1.0e - 7 \\ q_h &= 1.0e - 10 \\ q_r &= 1.0e - 12. \end{aligned} \tag{5.8}$$

The ensemble of particles $\mathbf{x}_i^j, i \in [1, k]$ is generated at each time j step following the modified equation (containing a deterministic nudging term, cf.(3.18))

$$\mathbf{x}_i^j = f(\mathbf{x}_i^{j-1}) + \hat{\boldsymbol{\beta}}_i^j + \mathbf{K}(\mathbf{d}^n - \mathbf{H} f(\mathbf{x}_i^{j-1})), \tag{5.9}$$

where $\mathbf{H} \in \mathbb{R}^{3m \times 3N}$ is the observation operator that converts the state vector space into the observational vector space and where

$$\hat{\boldsymbol{\beta}}_i^j \sim \mathcal{N}(\mathbf{0}, \hat{\mathbf{Q}}) \in \mathbb{R}^{3N} \tag{5.10}$$

is the perturbation vector added to the model at each integration time step. $\hat{\mathbf{Q}}$ is the error covariance matrix of the nudged model as it is generated at each time step according to the importance density. We chose the Gaussian perturbation vector such that

$$\hat{\mathbf{Q}} = \mathbf{Q} \quad \text{and} \quad \hat{\boldsymbol{\beta}}_i^j \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}). \tag{5.11}$$

This choice follows mainly, at first, a consideration of simplicity. In this sense, the truth run is just one realization of the ensemble of particles. Since the truth is unknown, it is possible to choose deliberately another stochastic realization for the truth run.

Observations

The observations are generated from the true trajectory at every assimilation time step by adding a Gaussian random error to it. The number of integration time steps between two assimilation times is $L = 100$. The observation error covariance matrix is diagonal. Since in our setting, the number m of observations is the same for each dynamical variable, the observation error covariance matrix is specified by $\boldsymbol{\gamma} \in \mathbb{R}^{3m}$, $\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, where

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_u & 0 & 0 \\ 0 & \mathbf{R}_h & 0 \\ 0 & 0 & \mathbf{R}_r \end{bmatrix} \in \mathbb{R}^{3m \times 3m}, \quad \mathbf{R}_u = r_u \mathbf{I}_m, \quad \mathbf{R}_h = r_h \mathbf{I}_m, \quad \mathbf{R}_r = r_r \mathbf{I}_m. \tag{5.12}$$

As numerical values, we choose

$$\begin{aligned} r_u &= 1.0e - 6 \\ r_h &= 1.0e - 5 \\ r_r &= 1.0e - 6. \end{aligned} \tag{5.13}$$

Since observation locations are the same for each variable, the observation operator has dimension $\mathbf{H} \in \mathbb{R}^{3m \times 3N}$, for a $3m$-dimensional observation vector.

Nudging matrix and parameter

The nudging matrix $\mathbf{K}$ has been used, for simplicity, either as

$$\mathbf{K} = K \mathbf{H}^T \mathbf{I}_{3m} \in \mathbb{R}^{3N \times 3m}, \tag{5.14}$$

where $K \in [0, 1]$ is a constant and arbitrarily chosen nudging parameter, or according to the need of maintaining dynamical balances, as in [1], choice which is more adapted to a 'spatially smooth' nudging, i.e. a nudging where the surrounding grid points of the particles are nudged as well. In that second case, we chose the matrix to be

$$\mathbf{K} = \tilde{K}(\mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1}) \in \mathbb{R}^{3N \times 3m}, \tag{5.15}$$

where $\tilde{K} \in \mathbb{R}$ is another nudging parameter than K. $\tilde{K}$ has to be chosen in such a way that the form of the nudging matrix corresponds to the Kalman gain matrix, where the analysis error covariance matrix $\mathbf{P}^a$ is replaced by the model error covariance matrix $\mathbf{Q}$ (cf. [26]). Not only does a Kalman gain matrix represent an 'optimal relaxation' of the state vectors in terms of resulting smallest possible analysis error, but in this way it is also ensured that the correspondence between the orders of magnitudes of observation and state vector is maintained, as well as physical dimensions [2]. This form of nudging matrix normalizes the increment $(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}^j))$ by the observation error covariance and nudges the evolved state vector by a magnitude that corresponds to the model error. With our use of the model error covariance $\mathbf{Q}$ as in (5.7), the nudging term in (5.9) only corrects u, h and r with observations of u, h and r respectively.

Physical parameter		Physical model
H_c	90.02 m	water level threshold for increased buoyancy
H_r	90.4m	water level threshold for rain production
ν_u	25000 m^2/s	diffusion constant for wind velocity
ν_h	25000 m^2/s	diffusion constant for water height
ν_r	200 m^2/s	diffusion constant for rain
Φ_c	899.77 m^2/s^2	geopotential
α	$2.5e^{-4}/s$	rain persistance factor
β	3.0	rain formation factor
nx	250 km	domain length
dx	500 m	spatial resolution
dt	5 s	integration time step
q_u	$1.0e^{-7}m/s$	truth model error variance for wind
q_h	$1.0e^{-10}m$	truth model error variance for water height
q_r	$1.0e^{-12}m$	truth model error variance for rain

Assimilation parameter		Ensemble run
q_u	$1.0e^{-7}$ m/s	model error variance for wind
q_h	$1.0e^{-10}$ m	model error variance for water height
q_r	$1.0e^{-12}$ m	model error variance for rain
r_u	$1.0e^{-6}$ m/s	observation error variance for wind
r_h	$1.0e^{-5}$ m	observation error variance for water height
r_r	$1.0e^{-6}$ m	observation error variance for rain
N	500	number of grid points
L	100	time steps per assimilation cycle
nsub	10.000	number of integration time steps
k	20	ensemble size

Table 5.1: Physical and assimilation parameter settings (cf. [59]).

5.3 Results

This chapter presents the experimental results of the particle filter assimilation tested on the modified shallow water model [59], and measured in terms Root Mean Square (RMS) error of the particles with respect to the synthetic true run. We measure the efficiency of the particle filter by comparing the RMS error of the filter with the RMS error of a nudged ensemble of particles that do not undergo resampling. Ideally, filtering would lead to a minimal error in all three variables. In the following sections, we show that this goal can only be partly achieved. For instance, a minimal RMS error for wind velocity does not necessarily correspond to a minimal RMS error of the amount of rain or of the water height level. This is why the problem setting is the following: What are the conditions unde which wind velocity, water height and rain attain their minimum RMS error, either separately or each in a separate combination with one of the others?

In what follows, we make essentially three types of tests. First, we test the impact of reducing the number of assimilated variables. Second, we reduce the number of observations available on the domain. Third, we use different types of nudging matrices. Overall, our goal was to understand the impact of a decreasing amount of information (choice and number of assimilated variables) on the results, while using a minimal number of ensemble members for the tests. We assimilate either all variables u, h and r, or two variables u and r, or only wind velocity u. By changing the number of assimilated variables among three present dynamical variables of the system, we intended to test the performance of the filter. Since radars measure wind velocity and rain reflectivity, we chose wind and rain as two-variable-combination for the assimilation tests with the efficient particle filter. In the case of a single assimilated variable, we chose the wind velocity. In the efficient particle filter, only a variable which undergoes stochastic perturbation can be assimilated. Since we wanted to lean on the original design of this toy model [59], in which the dynamics of the whole system is generated by small noise in the wind field only, the wind variable was a natural candidate. However, rain could have been chosen as well.

In Section 5.3.1, we first show how the model behaves without assimilation, taking 'model snapshots' at chosen assimilation time steps. We then use the most simple experiment set up and compare the 'free run' to the (unreal) case where observations are available at every grid point and where the nudging matrix $\mathbf{K}$ is diagonal. A first test of the efficient particle filter using some arbitrary nudging intensity showed a reduced RMS error compared to a nudged ensemble without resampling. But using another nudging parameter revealed no improvement or even a worse result of the filter. It turned out that an improvement of the filter happens only for a certain range of nudging parameters $K \in [0, 1)$, and that otherwise either the filter and the nudged ensemble give indistiguishable results or that the filter does worse than its nudged counterpart. A more complex reaction of the filter to nudging than initially thought led us to lead further investigations, from now on, only in terms of RMS error as a function of the nudging parameter.

In Section 5.3.2, we study a more realistic situation where observations are not available at every grid point. We choose a 'scarce observation operator' **H** which uniformly 'places' observations at 10, or 40 percent of the $N = 500$ grid points.

In Section 5.3.3, we show that the choice of the nudging matrix **K** has an impact on the results of the filter. In our experiments, a naively chosen diagonal nudging matrix can lead to better results than the nudging matrix proposed by [1].

5.3.1 Preliminary tests

In this subsection, we show results of first assimilation tests using the simplest framework, where observations are available at every grid point ($\mathbf{H} = \mathbf{I}$) and where the nudging matrix **K** is diagonal and of arbitrary magnitude ($\mathbf{K} = K\mathbf{I}$, $K \in [0, 1]$). This was done to get a first rough idea of the filter's behaviour. For the validation, we chose the RMS error defined with respect to the true model $\mathbf{x}_{tr}$ as

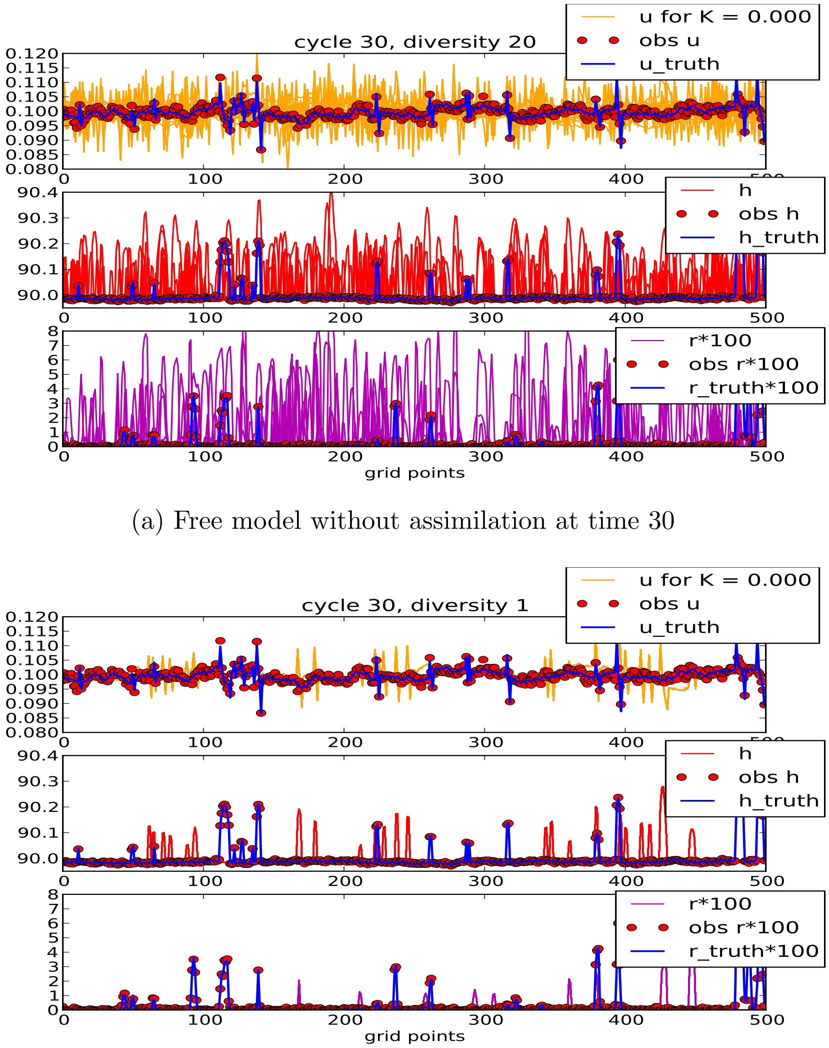
$$RMS = \sqrt{\overline{(\mathbf{x}_i - \mathbf{x}_{tr})^T (\mathbf{x}_i - \mathbf{x}_{tr})}}, \tag{5.16}$$

where the mean, indicated by a bar, is taken over the k ensemble members $\mathbf{x}_i (i \in [1, k])$, and compare it to the situation of nudging without resampling. In addition, we compute the spread or standard deviation of the ensemble, defined by

$$\sigma(\{\mathbf{x}_i\}) = \sqrt{\overline{(\mathbf{x}_i - \overline{\mathbf{x}})^2}}, \tag{5.17}$$

where $\overline{\mathbf{x}}$ is the mean state vector over the ensemble. This provides an indication of the variability of ensemble members. Ideally, the spread would be equivalent to the RMS error. In the following experiments, the RMS error and the spread are computed at each assimilation time step for a fixed nudging parameter K. They are then averaged, respectively, over 70 percent of the last assimilation time steps in order to give the model time to develop characteristic features of convection such as rain. In addition, they are both normalized by the value of the RMS error of the first assimilation time step. The graphs of the RMS error or spread per nudging were made after 4 repetitions of such an experiment.

Applying the SIR filter



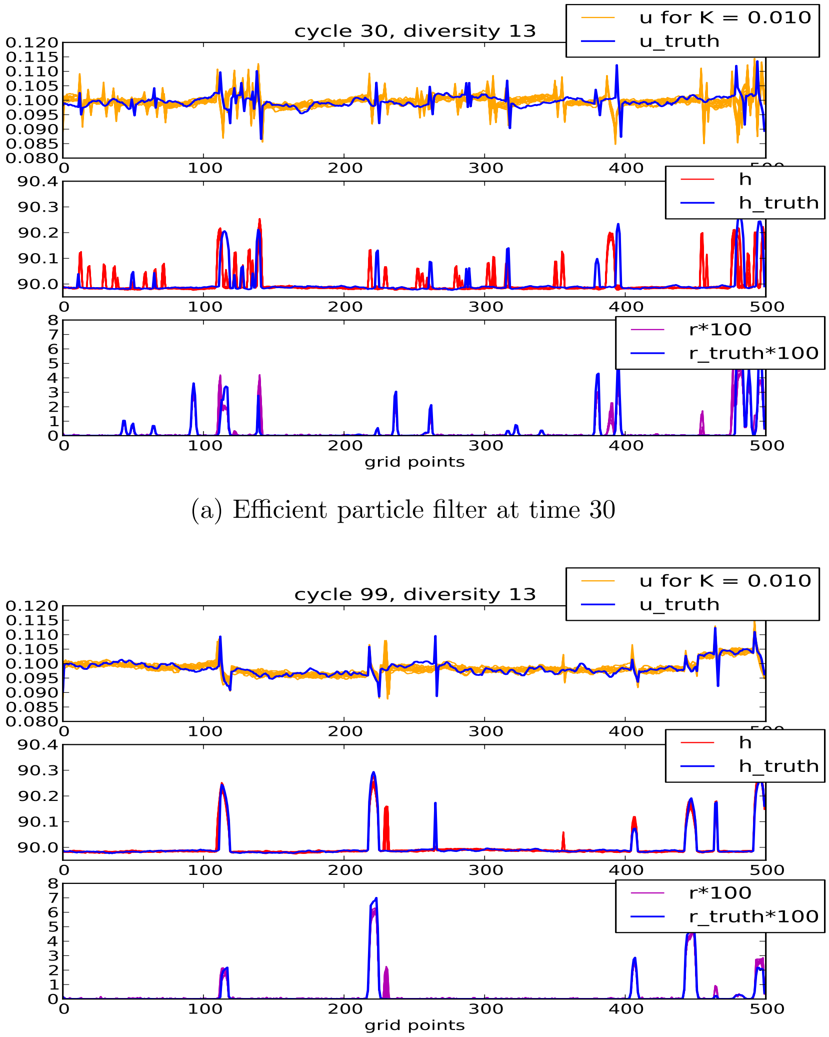
(a) Free model without assimilation at time 30

(b) SIR filter at time 30

Figure 5.1: Model state at assimilation time step 30 for an ensemble of 20 members. The upper panel (a) shows the free model, and the lower panel (b) the model after assimilation of all three variables with the SIR filter. All particles of wind velocity ($\mathbf{u}$), water height ($\mathbf{h}$) and rain ($\mathbf{r}$) are represented by yellow, red and magenta lines respectively. The observations, available at every grid point ($\mathbf{H} = \mathbf{I}$), are indicated by red dots. The SIR filter reduces the 'diversity' of the ensemble, i.e. the number of different members at given time, to 1, leading to filter degeneracy.

Figure 5.1 shows an ensemble of $k = 20$ free evolving particles together with the truth-run at assimilation time step 30 ('Free' model without any assimilation, panel a)), compared to a SIR assimilation (in panel b)). The SIR filter shows a typical behavior of filter collapse, since only one particle is replicated. It poorly captures the truth in all dynamical variables and generates erroneous features in wind, water height and rain.

Applying the particle filter

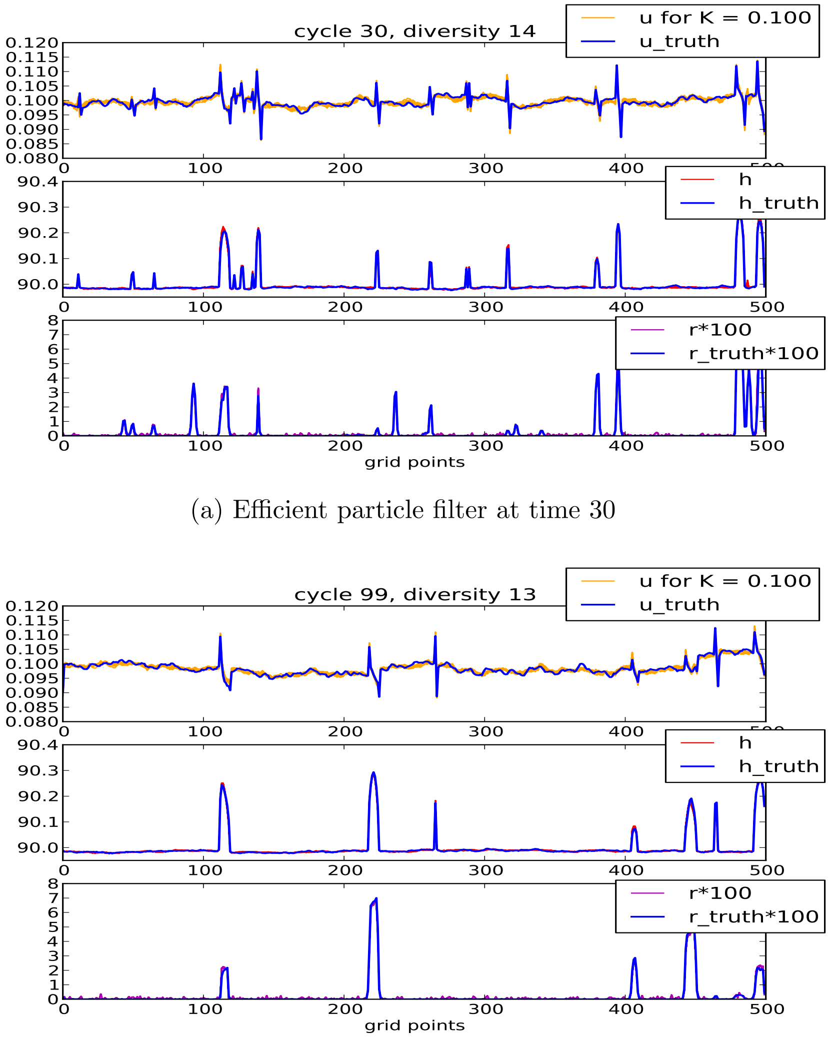


(a) Efficient particle filter at time 30

(b) Efficient particle filter at time 99

Figure 5.2: Wind velocity, water height and rain at assimilation time step 30 (a) and time step 99 (b) after applying the efficient particle filter for a 20 member ensemble with *weak nudging* $K = 0.01$.

Applying the efficient particle filter with a weak nudging forcing of $K = 0.01$ (Figure 5.2) shows a notable improvement of the ensemble's behaviour, compared to the SIR filter. Although not perfect, the efficient particle filter manages to capture the main features of wind disturbances, water height and rain. The truth is better reproduced by the filter after 99 time steps, than after 30 time steps where there are still many spurious features, indicating that a certain time is necessary to find the true solution. In Figure 5.2, the observations were omitted for a better visibility of the ensemble. Increasing the nudging strength in the present case, where observations are available at every grid point, makes the model perfectly adapt to the truth in all dynamical variables (Figure 5.3).

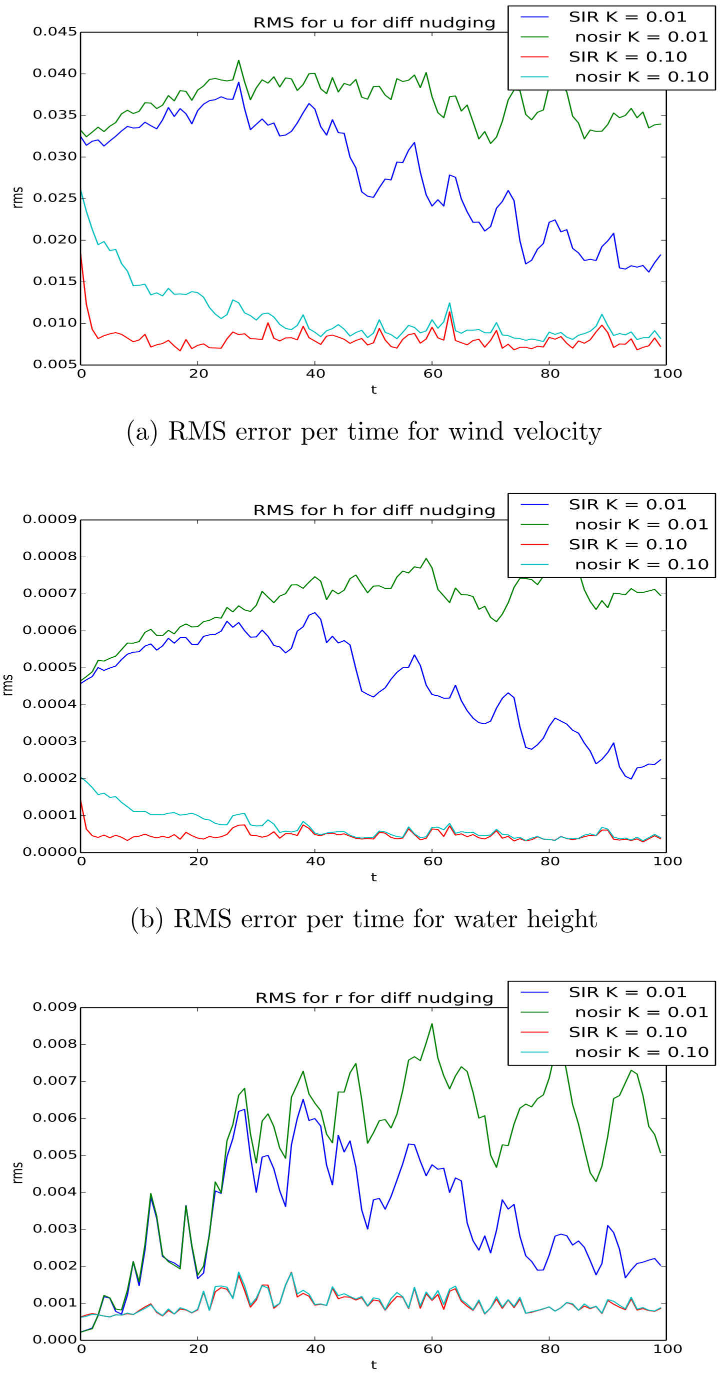


(a) Efficient particle filter at time 30

(b) Efficient particle filter at time 99

Figure 5.3: Wind velocity, water height and rain at assimilation time step 30 (a) and time step 99 (b) after applying the efficient particle filter for a 20 member ensemble with *stronger nudging* $K = 0.1$.

The stronger nudging is responsible for the ensemble matching the truth earlier, since the observational error is very small. These tests, though, just reflect a very idealized setting which is never available in 'real life' data assimilation, where observations are scarce in time as well as in space. The question remains how to measure the efficiency of the filter, with respect to other methods, and how it behaves in less ideal situations. One way of 'validating' the filter consists of comparing its RMS error per time with the RMS error per time of an ensemble of particles, which are nudged in the same way as in the filter but without undergoing the resampling step at each assimilation time.



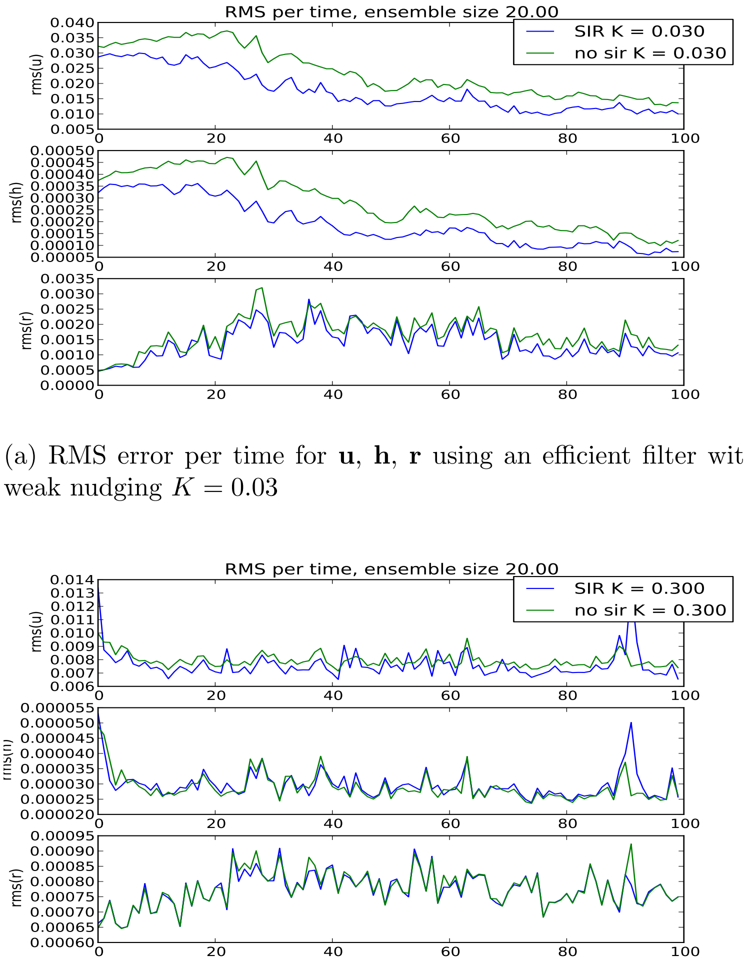
(a) RMS error per time for wind velocity

(b) RMS error per time for water height

(c) RMS error per time for rain

Figure 5.4: RMS error per time of **u** (picture a), **h** and **r** (b and c) of the efficient particle filter (blue and red lines) vs a nudged model (green and turquoise lines) without resampling, for an ensemble of 20 members. The resampling procedure, denoted by 'SIR' in the legend, induces a difference in the RMS, between the filter and the regime of mere nudging, only for the weaker nudging parameter $K = 0.01$. After 100 time steps, there is no distinguishable difference anymore between filter and nudged ensemble, for a stronger nudging of $K = 0.1$.

Figures 5.4 and 5.5 show that, although - absolutely speaking - the smallest error is achieved in time with a stronger nudging coefficient K, the resampling procedure does not enable the filter to outperform the ensemble of nudged particles anymore. In other words, there seems to be a dependency on the nudging parameter which decides whether the filter outpasses a nudged ensemble, or not. Upon this difficulty, the question arises if it is worth using the filter if nudging particles only gives the best result under sufficiently strong nudging. In what follows, the RMS error is plotted in dependency of the nudging parameter K.

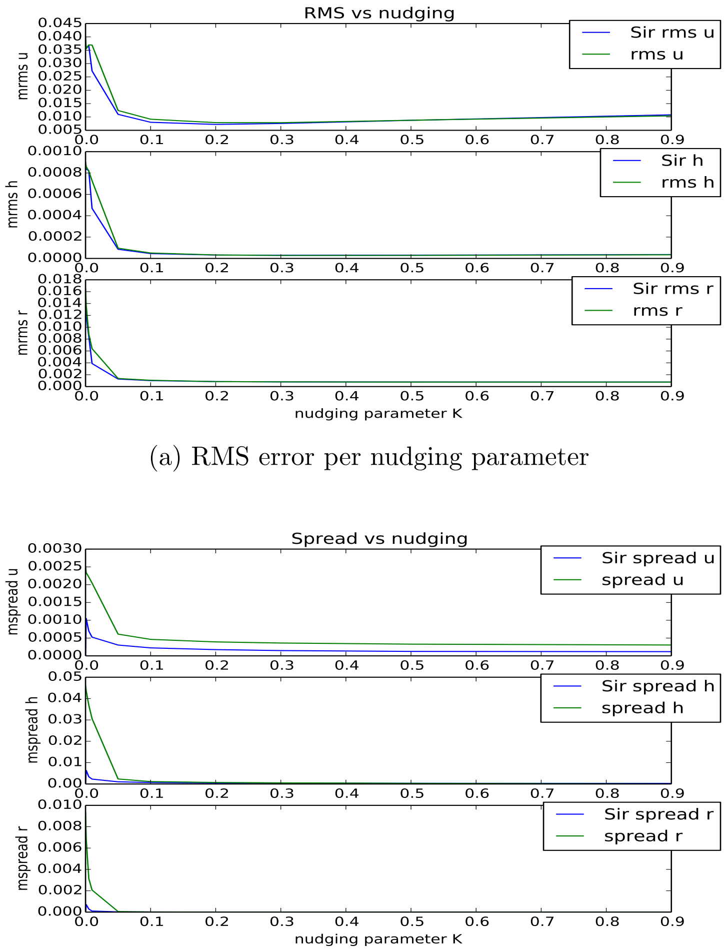


(a) RMS error per time for **u**, **h**, **r** using an efficient filter with weak nudging $K = 0.03$

(b) RMS error per time using an efficient particle filter using stronger nudging $K = 0.3$

Figure 5.5: RMS error per time of **u** (top picture), **h** and **r** (bottom picture) for the efficient particle filter (blue line) vs nudging of the model (green line) only, for a 20 member ensemble. The resampling procedure reduces the error compared to mere nudging only for certain nudging parameters.

Assimilation of three variables



(a) RMS error per nudging parameter

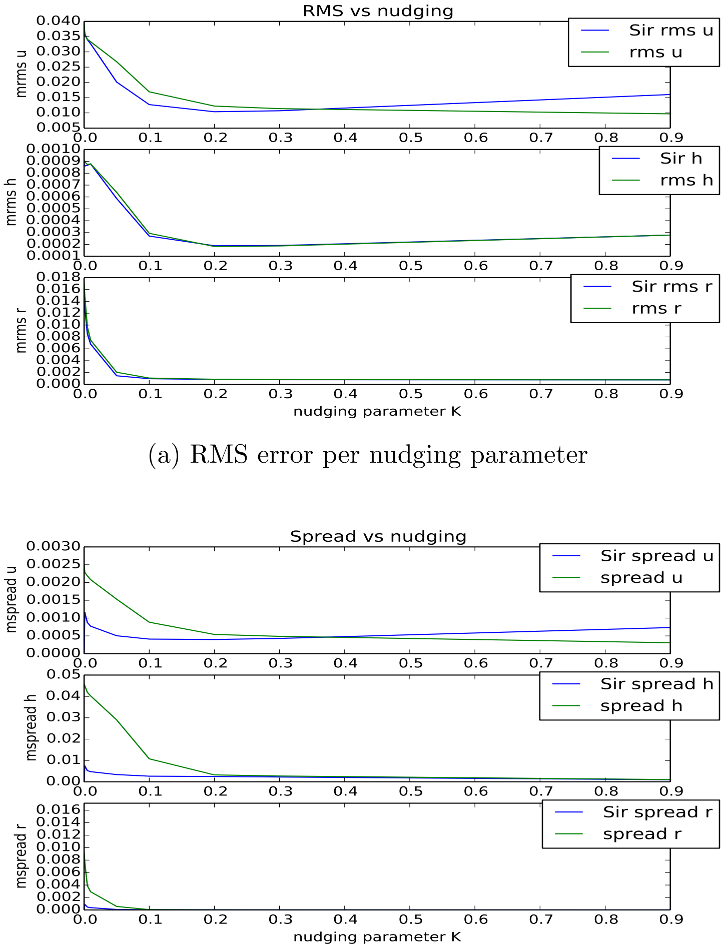
(b) Spread per nudging parameter

Figure 5.6: Mean RMS error (a) and spread (b) per nudging parameter of the efficient particle filter (blue line), compared to the nudged ensemble (green line), for an ensemble of 20 members. For a fixed nudging parameter, the drawn RMS error value corresponds to the mean RMS error and mean spread taken over the last 70 (assimilation) time steps.

Figure (5.6) shows the mean RMS error of **u**, **h**, and **r** respectively (taken over 70 assimilation time steps) as a function of the nudging intensity K. Only for small values of nudging ($K < 0.05$), there is a difference between the filter and the nudged ensemble.

Assimilating two variables: wind velocity and rain

Assimilating only two variables (Figure 5.7), or one variable (Figure 5.8), does not improve the efficiency of the filter. Only in the wind field, the difference of RMS errors between filter and nudged ensemble differs. Nudging more becomes more penalizing for the filter in the wind velocity. There is no difference in the variables **h** and **r**, in terms of RMS, between the filter and the nudged ensemble.

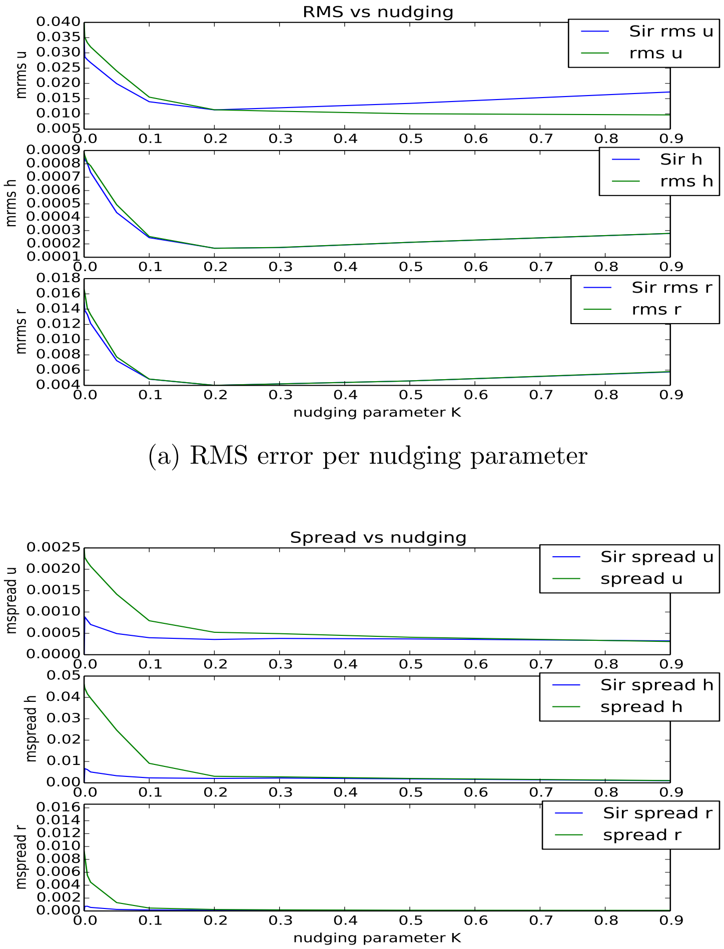


(a) RMS error per nudging parameter

(b) Spread per nudging parameter

Figure 5.7: RMS error (a) and spread (b) per nudging parameter of the efficient particle filter (blue line), compared to the nudged ensemble (green line), for the assimilation of the two variables wind and rain, with an ensemble of 20 members. The filter does not outperform a nudged ensemble without resampling.

Assimilating wind velocity



(a) RMS error per nudging parameter

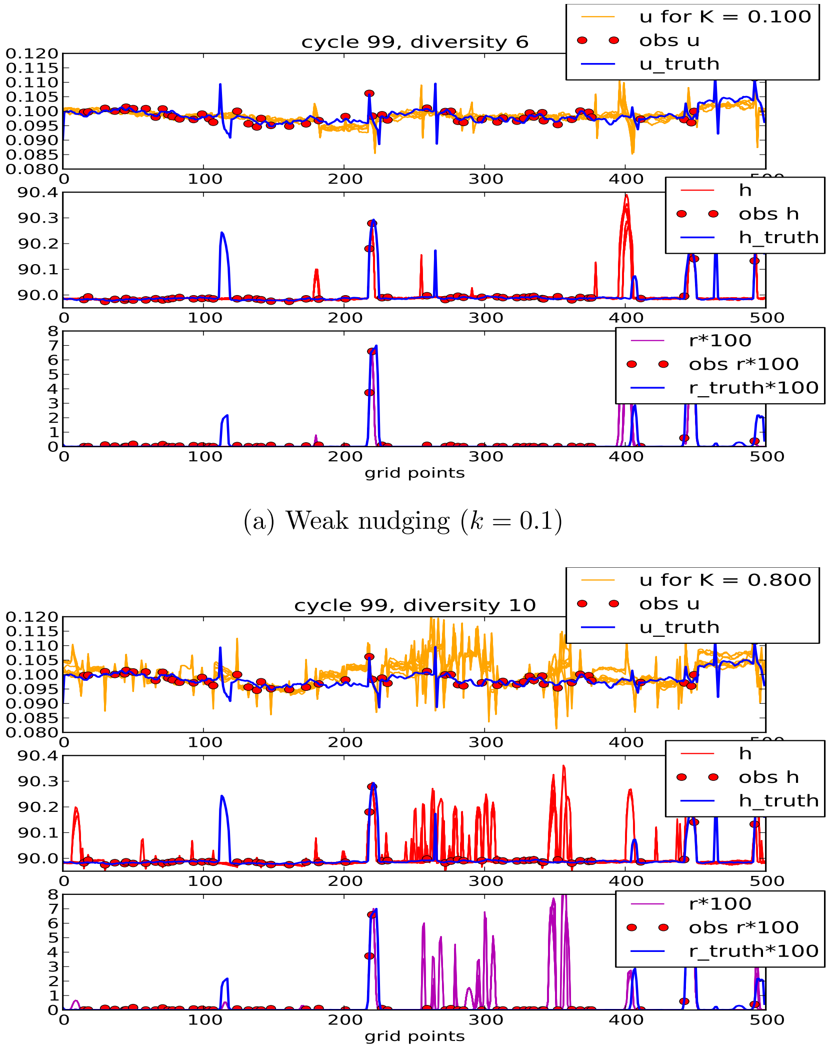
(b) Spread per nudging parameter

Figure 5.8: RMS error (a) and spread (b) per nudging parameter of the efficient particle filter (blue line), compared to the nudged ensemble (green line), for the assimilation of one variable, the wind velocity, with an ensemble of 20 members.

The 'poor' performance of the filter compared to the nudged ensemble is to be expected in the case of $\mathbf{H} = \mathbf{I}$, given that nudging particles towards observations available everywhere just consists of replacing the particles by the observations. These tests were made just to get a first rough idea of the filter's behaviour.

5.3.2 Scarce observations

In this section, we examine the result of reducing the number of observations in space. We produce two examples, where either 10, or 40 percent of uniformly distributed observations among the 500 grid points are kept. The tests were made with a diagonal nudging matrix **K** for the assimilation of three (**u**, **h**, **r**), two (**u**,**r**) or only of one (**u**) variable.



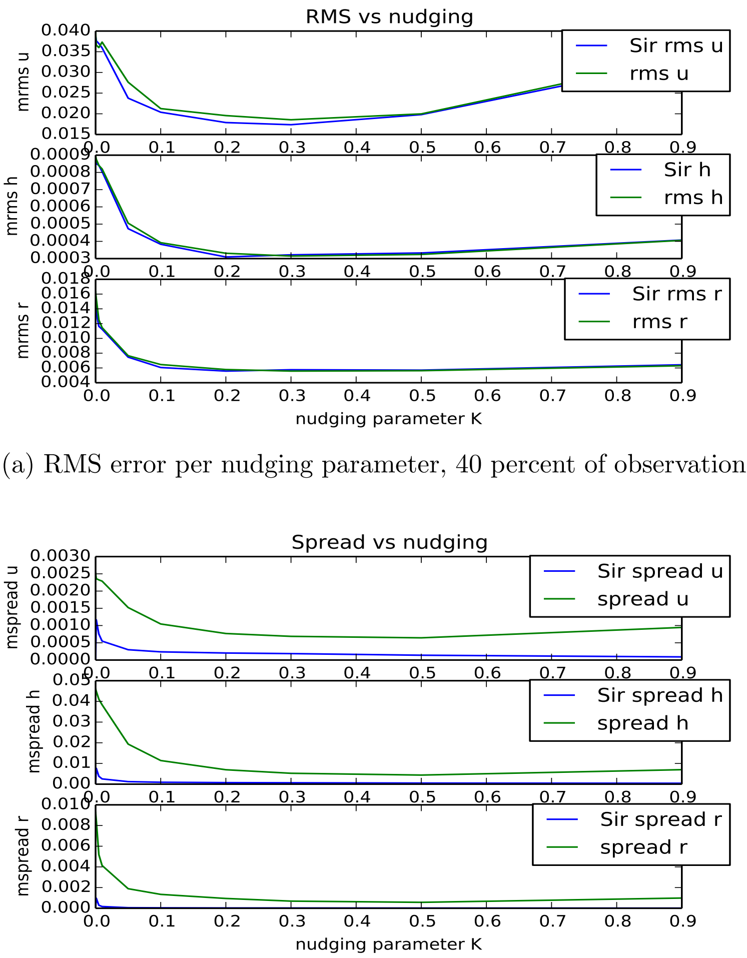
(a) Weak nudging ($k = 0.1$)

(b) Strong nudging ($k = 0.8$)

Figure 5.9: Model at time step 99. In this example, three variables are assimilated, with 10 percent of observations, represented by red dots, among the $N = 500$ grid points of the domain. The strong nudging (panel b)) creates imbalances that are transmitted to all fields.

Figure 5.9 shows a first example of model variables for a filter assimilating the three variables, for only 10 percent of spatially available observations. Reducing the number of observations in space augments the RMS error in all variables. It is expected that nudging in the present test configuration becomes penalizing for nudging parameters that surpass a maximal threshold value.

Assimilating three variables

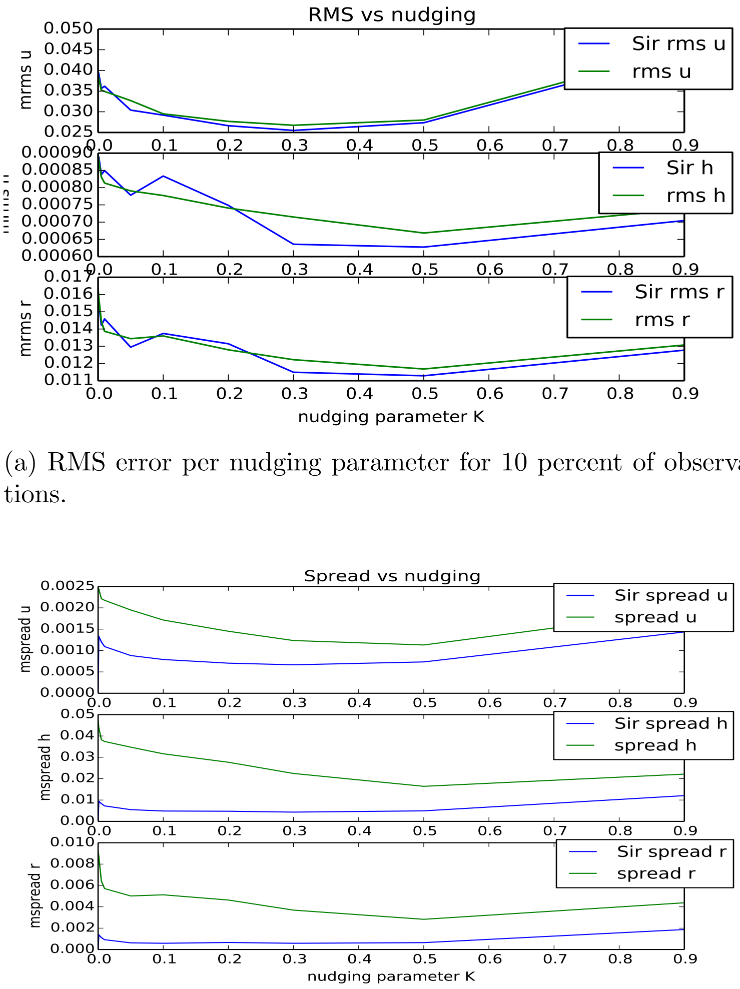


(a) RMS error per nudging parameter, 40 percent of observations

(b) Spread per nudging parameter, 40 percent of observations

Figure 5.10: Assimilation of 3 variables and 40 percent of observations for an ensemble of 20 members.

Figure 5.10 shows the comparative RMS error (panel a)) and spread (panel b)) per nudging parameter, between the filter (blue line) and the nudged model without resampling (green line), for an assimilation of all three variables and observations available at 40 per cent of the grid points. In comparison to Figure 5.6 where observations were available at all grid points, the smallest possible RMS error in Figure 5.10 that can be achieved here by all variables, becomes higher with fewer observations. At the same time, the RMS error curves of the three variables steepen compared to the case of $\mathbf{H} = \mathbf{I}$ with increasing nudging parameter, after a certain nudging threshold is reached. Nudging thus leads to faster growing errors and has a more penalizing effect when observations become scarce.



(a) RMS error per nudging parameter for 10 percent of observations.

(b) spread per nudging parameter for 10 percent of observations.

Figure 5.11: RMS error (panel a)) and spread (panel b)) per nudging parameter for ten percent of observations. The assimilation was done for 3 variables for an ensemble of 20 members.

Figure 5.11 shows the same test as before for ten percent of spatially available observations. The minimally achieved RMS error is again slightly higher than in the case of more dense (40 percent) observations. The filter outperforms the 'nudged-only' model with decreasing number of observations not only in the wind variable (cf. Figure 5.9), but in all the variables **u**, **h** and **r**. Hence, a distinct outperformance of the efficient particle filter compared to a regime with nudging only becomes evident, in all variables, by scarcening the available observations.

Assimilating two variables

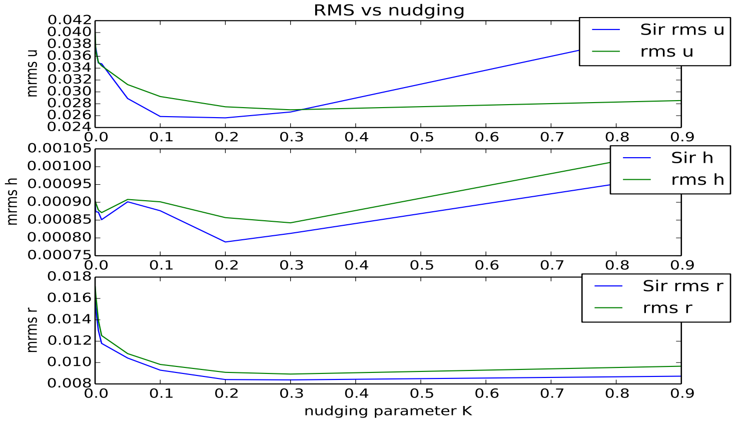


Figure 5.12: RMS error per nudging parameter for the assimilation of 2 variables and 40 percent of observations, for an ensemble of 20 members.

Figures 5.12 and 5.13 show the RMS error and the spread, per nudging parameter, for the assimilation of wind and rain for observations available at 40 percent of the domain. The filter leads to better results in all variables, compared to the nudged model.

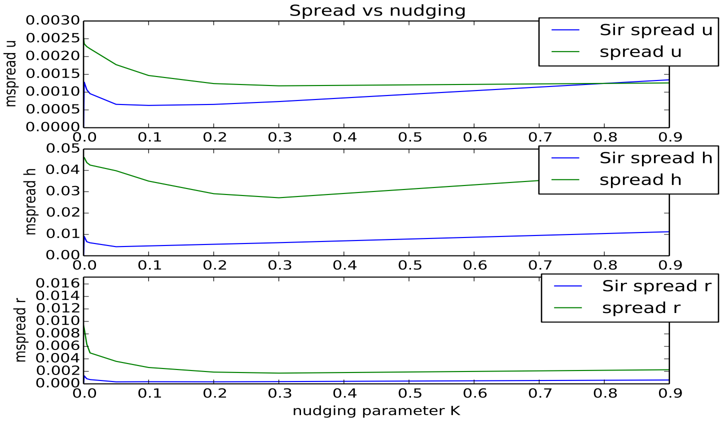


Figure 5.13: Spread per nudging parameter for the assimilation of 2 variables and 40 percent of observations, for an ensemble of 20 members.

When scarcening observations in space even more to 10 percent for the assimilation of two variables, the resulting RMS error curves become more complex (Figures 5.14 and 5.15).

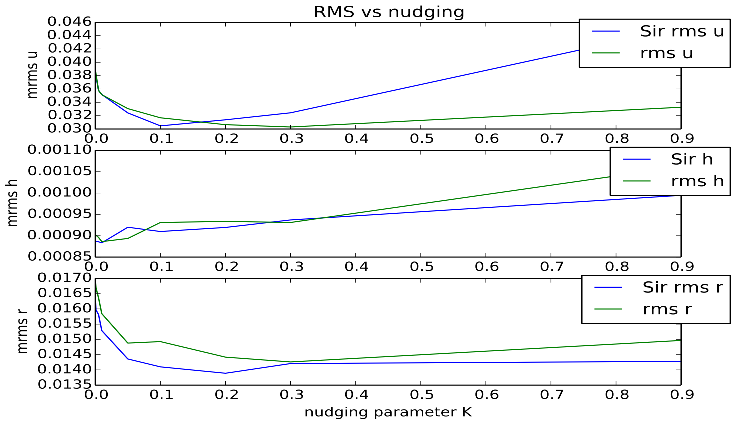


Figure 5.14: RMS per nudging parameter for the assimilation of 2 variables and 10 percent of observations, for an ensemble of 20 members.

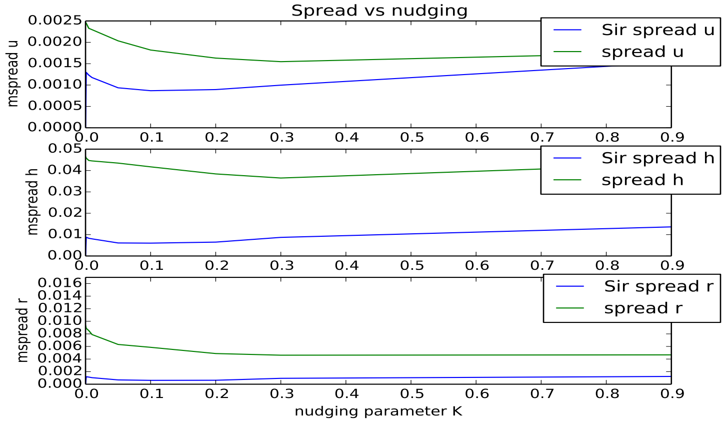
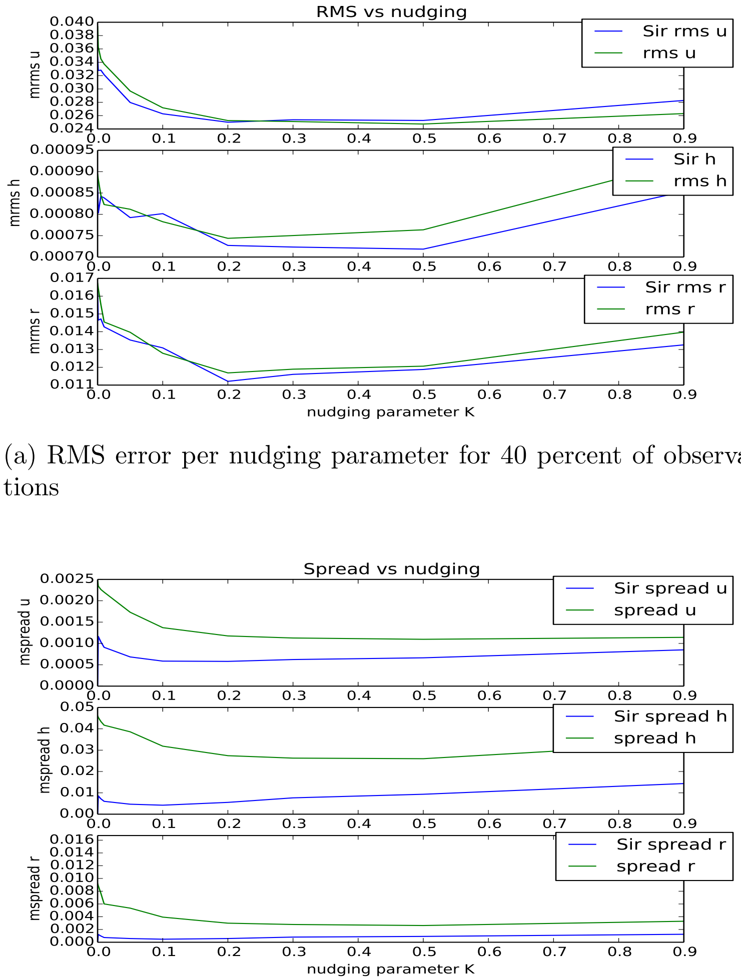


Figure 5.15: Spread per nudging parameter for the assimilation of 2 variables and 10 percent of observations, for an ensemble of 20 members.

While the filter and the nudged model are clearly distinguishable in all variables, they seem to be competitive in **u** and **h** for different ranges of the nudging parameter. The absolute minimum in the RMS error of **u** is obtained by the nudged model without resampling. This minimum occurs for a larger nudging parameter than the parameter corresponding to the filter's minimal RMS error. Regarding other variables, the filter performs better than nudging only. The minimum of each variable is achieved for different nudging parameters. The wind variable seems to be penalized stronger by nudging than other variables. Again as in Figure 5.10, the smallest possible RMS error that can be achieved (with varying nudging parameter) increases with diminishing the number of assimilated variables (here from 3 to 2).

Assimilating one variable

Figure 5.16 shows the RMS error and the spread curves per nudging parameter for the filter after assimilating the wind velocity, only. Resampling does not lead to an improvement of the minimal RMS error over the nudged model in the **u**-variable (the green and the blue line are nearly indistinguishable for a wide range of nudging parameters in the top panel a), it does in the other variables.

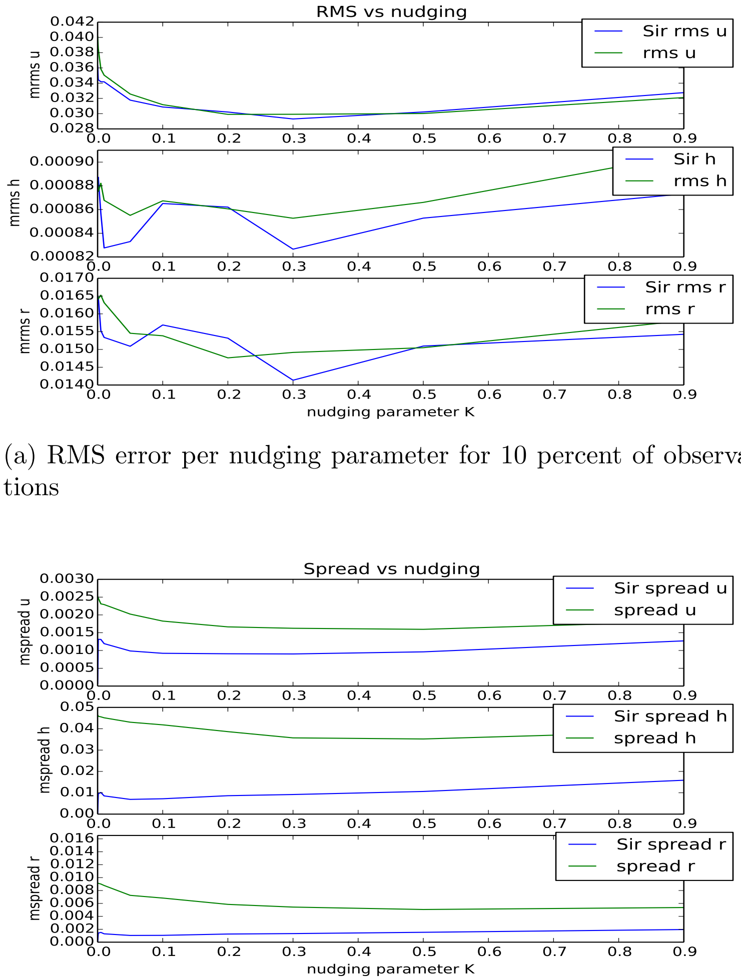


(a) RMS error per nudging parameter for 40 percent of observations

(b) Spread per nudging parameter for 40 percent of observations

Figure 5.16: Assimilation of 1 variable (wind velocity) and 40 percent of observations for an ensemble of 20 members.

A further reduction of observations to 10 percent (Figure 5.17) leads again to an increased minimal error in all variables, compared to assimilating more variables and to more dense observations. An improvement of the filter is barely given in the wind variable, but it is significant in the other variables **h** and **r**. The absolute minimum RMS error in all variables is obtained for the same nudging parameter $K = 0.3$.

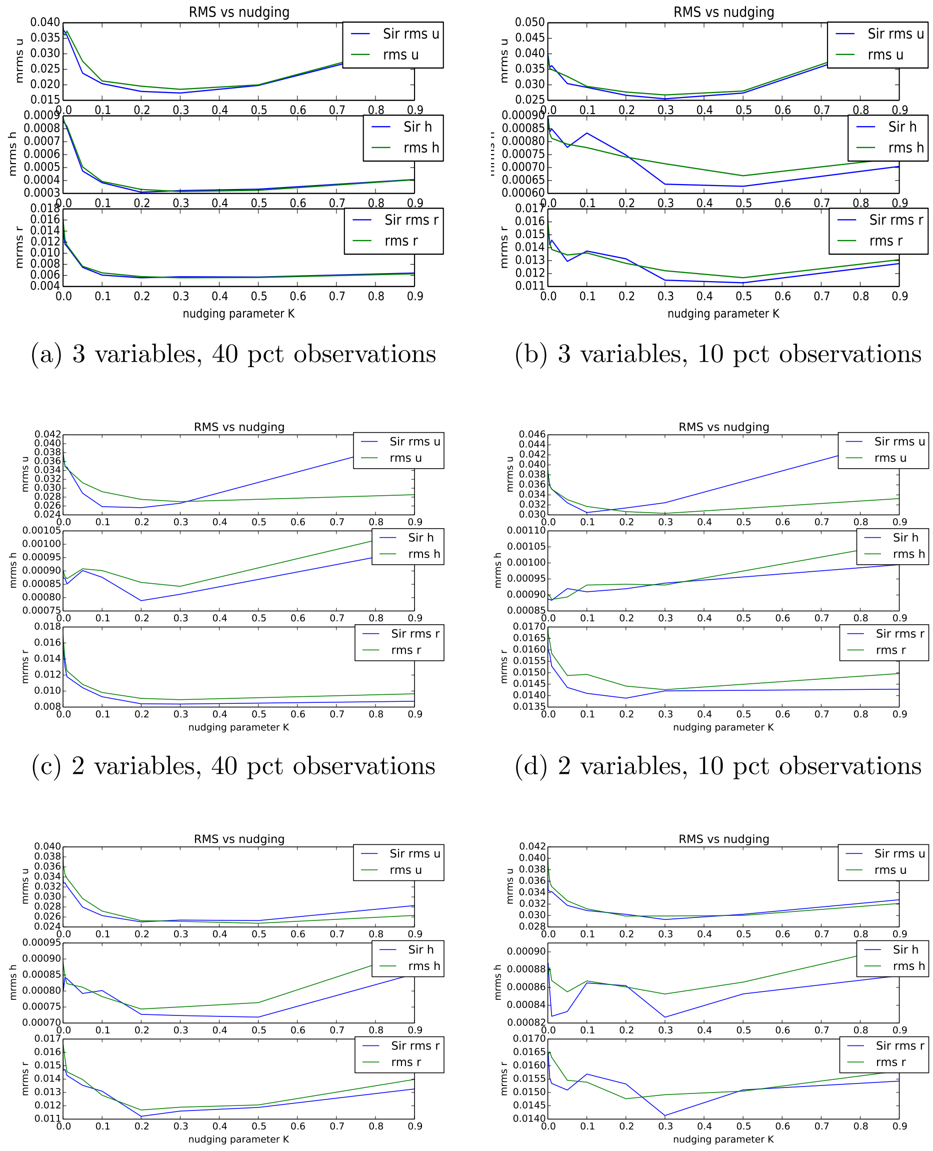


(a) RMS error per nudging parameter for 10 percent of observations

(b) Spread per nudging parameter for 10 percent of observations

Figure 5.17: Assimilation of 1 variable (wind velocity) and 10 percent of observations for an ensemble of 20 members.

A comparative figure of the 'best score' RMS error values of the different configurations treated in this section is represented in Figure 5.19. It shows that the more observations are available in space and the more variables of a system are assimilated, the smaller is the RMS error. It also illustrates that in the case of 100 percent of observations, the filter cannot do better than nudging, which was to be expected. For a better visibility of the previous results, we grouped all the plots above together in Figure 5.18.



(a) 3 variables, 40 pct observations

(b) 3 variables, 10 pct observations

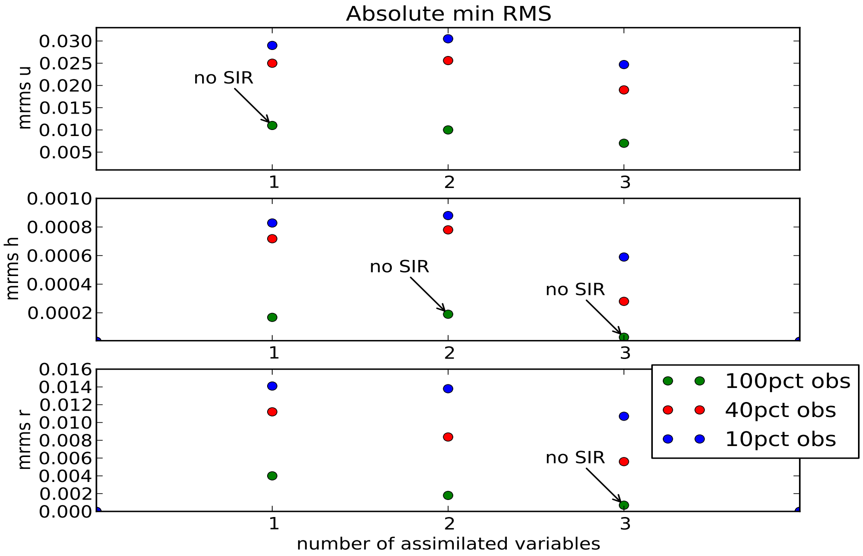
(c) 2 variables, 40 pct observations

(d) 2 variables, 10 pct observations

(e) 1 variable, 40 pct observations

(f) 1 variable, 10 pct observations

Figure 5.18: Comparison of the evolution of the RMS error per nudging parameter k for the assimilation of 3 (panels a) and b)), 2 (panels c) and d)) variables and 1 variable (panels e) and f)), for 40 percent (left column) vs 10 percent (right column) of available observations.



(a) Minimal RMS error for the efficient particle filter using a diagonal **K**-matrix

Figure 5.19: Absolute minimal mean RMS error for **u** (top figure), **h**, and **r** (bottom figure) per number of assimilated variables. This mean corresponds to a mean RMS error of the model variable taken over the last 70 timesteps, mean taken again over 5 realizations of the experiment. The arrow with 'no SIR' indicates that in this experiment, the nudged-only ensemble led to a smaller RMS error than the efficient particle filter.

5.3.3 Impact of the nudging matrix

Non-diagonal nudging matrix K

For the non-diagonal version of the nudging matrix $\mathbf{K} \in \mathbb{R}^{\mathbf{3N\times3m}}$ defined in (5.15), we use

$$\begin{aligned}\mathbf{K} = \tilde{K}\mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1} &= \tilde{K}\underbrace{\begin{bmatrix}\mathbf{Q}_u & 0 & 0\\ 0 & \mathbf{Q}_h & 0\\ 0 & 0 & \mathbf{Q}_r\end{bmatrix}}_{3N\times3N}\underbrace{\begin{bmatrix}\mathbf{H}_u & 0 & 0\\ 0 & \mathbf{H}_h & 0\\ 0 & 0 & \mathbf{H}_r\end{bmatrix}^T}_{3N\times3m}\underbrace{\begin{bmatrix}\mathbf{R}_u^{-1} & 0 & 0\\ 0 & \mathbf{R}_h^{-1} & 0\\ 0 & 0 & \mathbf{R}_r^{-1}\end{bmatrix}}_{3m\times3m}\\ &= \tilde{K}\begin{bmatrix}\mathbf{Q}_u\mathbf{H}_u\mathbf{R}_u^{-1} & 0 & 0\\ 0 & \mathbf{Q}_h\mathbf{H}_h\mathbf{R}_h^{-1} & 0\\ 0 & 0 & \mathbf{Q}_r\mathbf{H}_r\mathbf{R}_r^{-1}\end{bmatrix}\\ &= \tilde{K}\begin{bmatrix}\mathbf{K}_u & 0 & 0\\ 0 & \mathbf{K}_h & 0\\ 0 & 0 & \mathbf{K}_r\end{bmatrix} \in \mathbb{R}^{3N\times3m}, \qquad (5.18)\end{aligned}$$

where **Q** is our model perturbation covariance matrix (5.7) with two off-diagonals, **R** the diagonal observation error covariance matrix (5.12), and $\tilde{K} \in \mathbb{R}$ a real parameter. The nudging matrix (5.18) introduces by construction a *different magnitude of relaxation for each* *dynamical variable.* In this, it substantially differs from the diagonal **K**-matrix in (5.14), where each variable was nudged by the same multiple of the distance between observation and particle.

In principle, any non-diagonal **K**-matrix (especially if **Q** was not block-diagonal) would generate nudging contributions in observed grid points coming from all (at most 3N if $m = N$) coordinates of the nudging vector $(\mathbf{d_x} - \mathbf{H}f(\mathbf{x}))$. These contributions add up within one coordinate of the nudging term $\mathbf{K}(\mathbf{d} - \mathbf{H}f)$, and they increase the total nudging magnitude compared to a diagonal matrix **Q**. In our case of a block-diagonal **Q**-matrix, only observational increments of few (at most five) coordinates of the same dynamical variable add up when applying the nudging matrix on a state vector. We are in a situation very close to a diagonal matrix, but with the difference to the former diagonal **K**-matrix used in the previous Section that now, the nudging intensity has a very different magnitude for each dynamical variable.

Also, when using our matrices **Q** and **R** specified in Section 5.2, the resulting submatrices of **K** for **u**, **h** and **r** contain one element per row that is of the order of magnitude $(K_u)_{ij} = 10^{-1}$, $(K_h)_{ij} = 10^{-5}$ and $(K_r)_{ij} = 10^{-6}$, respectively. Since all other elements per row are close to zero, this leads to a *much weaker relaxation* forcing than with a diagonal nudging matrix **K**, as used in the previous sections. Given the above non-diagonal **K**-matrix, the question is by which parameter $(\tilde{K})$ we can maximally multiply the non-diagonal nudging term in order to have a comparable nudging magnitude as for the diagonal **K**-matrix ? If we compare the elements' sums of the two **K**-matrices for $\mathbf{H_y} = \mathbf{I}_N$, $\mathbf{y} \in \{\mathbf{u}, \mathbf{h}, \mathbf{r}\}$, we have for a

$$\text{non-diagonal } \mathbf{K}\text{-matrix:} \quad \sum_{i,j} \mathbf{K}_{ij} \sim 50, \tag{5.19}$$

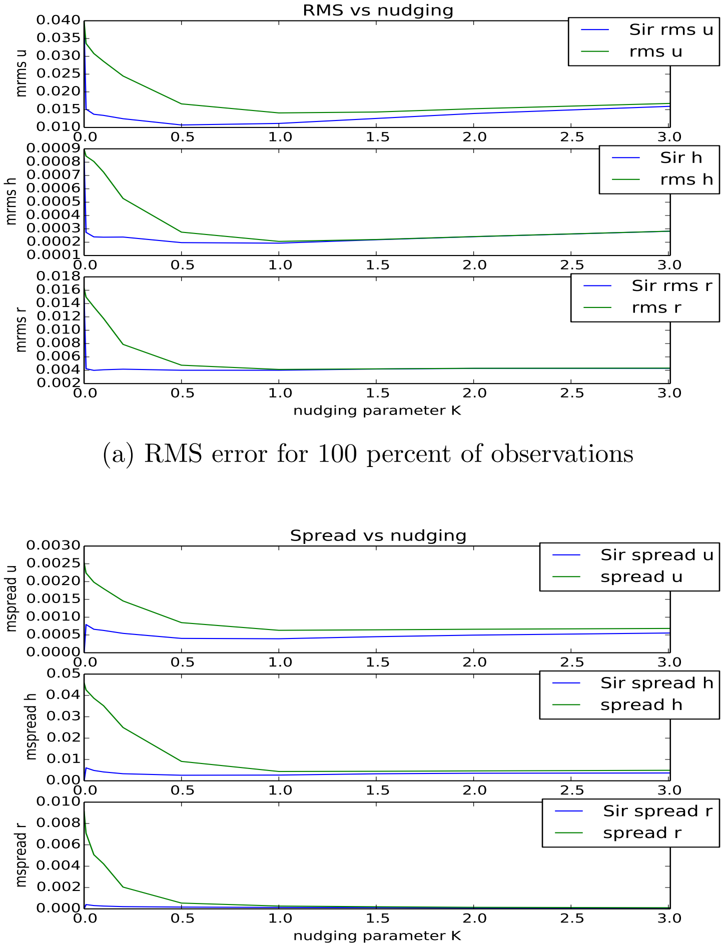
$$\text{diagonal } \mathbf{K}\text{-matrix:} \quad \sum_{i,j} \mathbf{K}_{ij} \sim 1500, \quad \text{if} \quad \mathbf{K_y} = \mathbf{I}_N, \quad \mathbf{y} \in \{\mathbf{u}, \mathbf{h}, \mathbf{r}\}. \tag{5.20}$$

The **h**- and **r**-contributions to the total **K**-matrix in (5.19) are negligible (of the order $K_u \sim 10^{-4}$ and $K_h \sim 10^{-5}$), whereas in the diagonal case, the contributions from the different variables are equal (and equal to 500 respectively). Hence, the nudging intensity of the variable **u** in the nondiagonal matrix is about 10 times smaller than in the diagonal one. Since 1.0 is the maximal nudging strength in the diagonal case, this means we are not allowed to nudge the **u** field more than 10 times stronger than we did in the diagonal case.

Beyond a nudging parameter of $K \sim 1.0$ in the case of the diagonal nudging matrix, the model integration cannot be performed numerically anymore. The RMS error starts growing again, after reaching its minimum, for nudging parameters larger than $\sim 0.3 - 0.5$. Therefore, we chose a nudging parameter $\tilde{K}$ in $\tilde{K}\mathbf{K}(\mathbf{d} - \mathbf{H}f)$ (cf. (5.15)) for the non-diagonal **K**-matrix which lies in the range $\tilde{K} \in [0, 3]$. Beyond the value of 3.0, nudging becomes obsolete as well.

Figure 5.20 shows the RMS error (panel a)) and the spread (panel b)) of all three variables for the efficient particle filter (blue line) versus a nudged ensemble (green line) without resampling, as a function of the nudging parameter (denoted by $K := \tilde{K}$ in the figure label), and for everywhere available observations. The minimal RMS error values in all variables are attained with the efficient filter, which largely surpasses the nudging-only regime until a nudging value of ~ 1.0. Beyond this nudging parameter, the filter equals the nudged ensemble. We notice that for $\mathbf{H} = \mathbf{I}$, the diagonal matrix (cf. Figure 5.6) gives better results than the non-diagonal one.

Coverage of 100% of observations for three assimilated variables

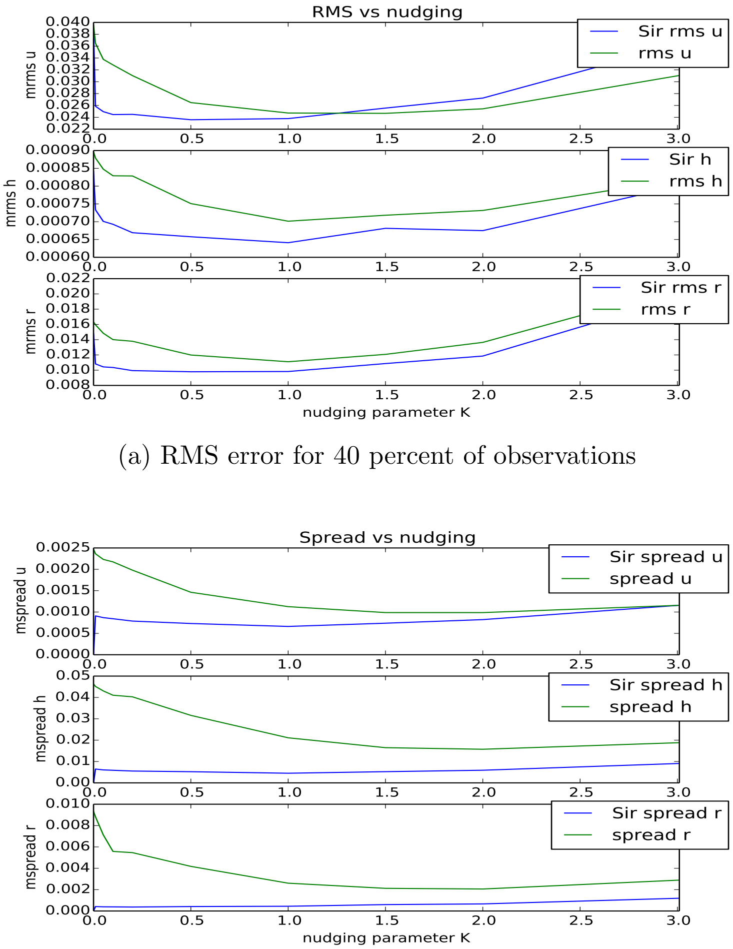


(a) RMS error for 100 percent of observations

(b) Spread for 100 percent of observations

Figure 5.20: Assimilation of 3 variables, RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of three variables, everywhere available observations and for a non-diagonal $\mathbf{K}$-matrix.

Coverage of 40% of observations for three assimilated variables



(a) RMS error for 40 percent of observations

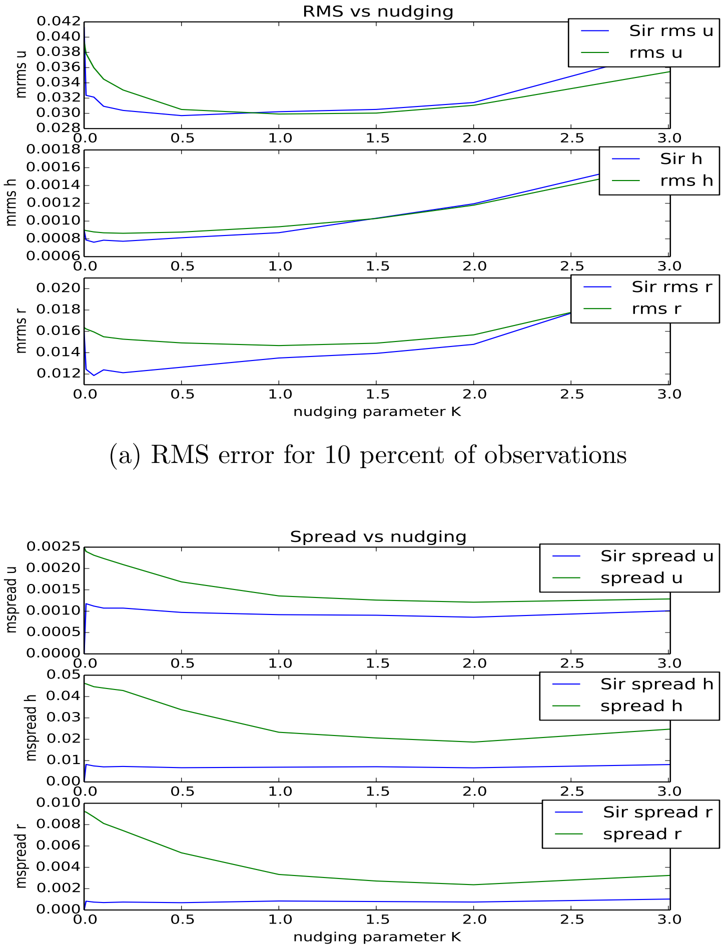
(b) Spread for 40 percent of observations

Figure 5.21: Assimilation of 3 variables, RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of three variables, for observations at 40 percent of the domain and for a non-diagonal $\mathbf{K}$-matrix.

When scarcening the observations in space from 100 to 40 percent of the domain (Figure 5.21), the minimal RMS errors of **u**, **h** and **r** increase at least by a factor 2. Simultaneously, nudging more than by the parameter corresponding to the minimum RMS error ($K = 1.0$ in Figure 5.21a)) leads to faster growing error in all variables than in the presence of observations everywhere.

Further decreasing the number of observations from 40 to 10 percent of the domain (Figure 5.22) increases again the absolute minimal RMS error and further reduces the limit nudging value above which the RMS error curve diverges. While in the case of 40 percent, the RMS error curve started to grow again beyond a nudging factor of 1.0, it already grows when nudging more than $K \approx 0.1$ in the case of 10 percent of observations. This is especially the case for $\mathbf{h}$ and $\mathbf{r}$.

Coverage of 10% of observations for three assimilated variables



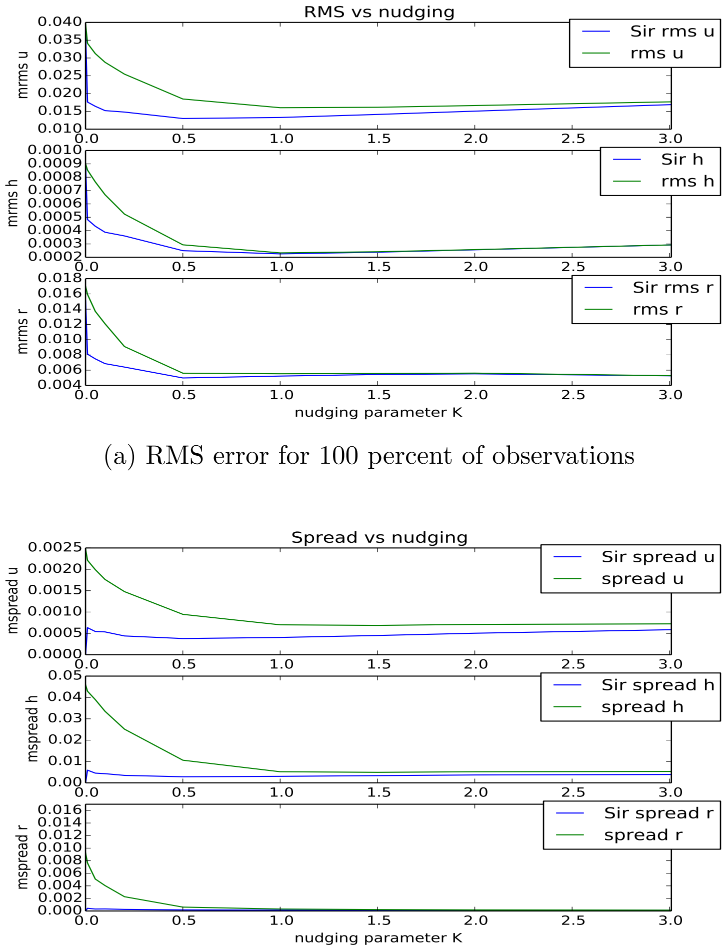
(a) RMS error for 10 percent of observations

(b) Spread for 10 percent of observations

Figure 5.22: Assimilation of 3 variables, RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of three variables, for observations at 10 percent of the domain and for a non-diagonal $\mathbf{K}$-matrix.

In Figures 5.23-5.25, where observations are reduced from 100 over 40 to 10 percent for the assimilation of wind and rain, the same phenomenon can be observed. For two assimilated variables, the reduction of observations from 40 to 10 percent leads to a shift of the minima of the RMS error curves (mainly in water height and rain) towards smaller nudging parameters (after a slight shift to higher nudging values when passing from 100 to 40 percent). This indicates that nudging is not allowed to be as strong in the presence of fewer observations.

Coverage of 100% of observations for two assimilated variables

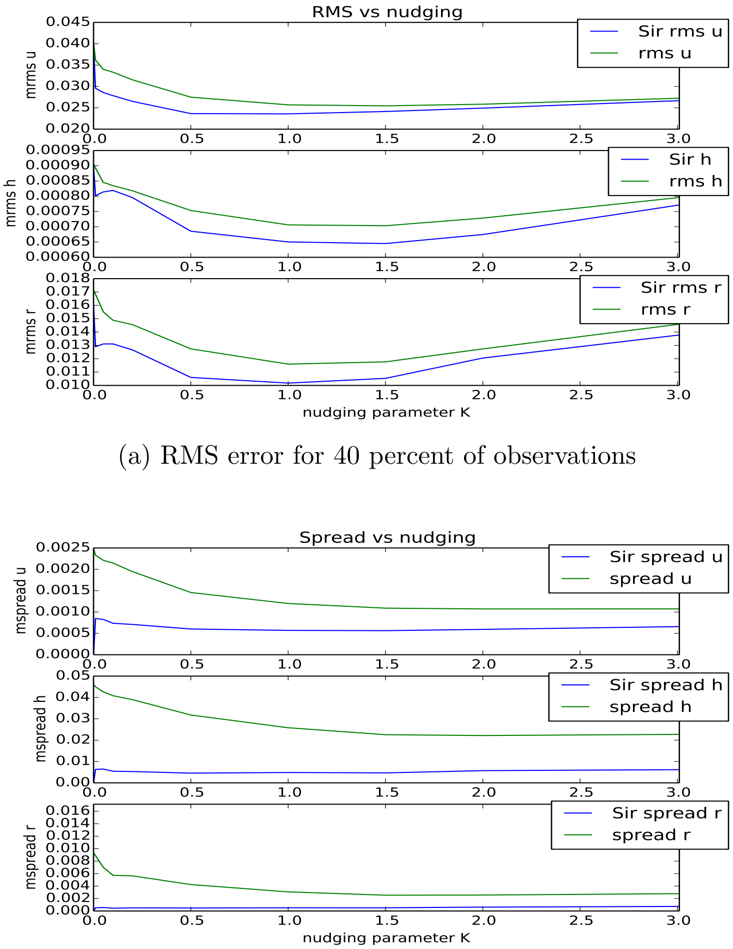


(a) RMS error for 100 percent of observations

(b) Spread for 100 percent of observations

Figure 5.23: Assimilation of 2 variables (wind and rain), RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of two variables, for everywhere available observations and for a non-diagonal $\mathbf{K}$-matrix.

Coverage of 40% of observations for two assimilated variables

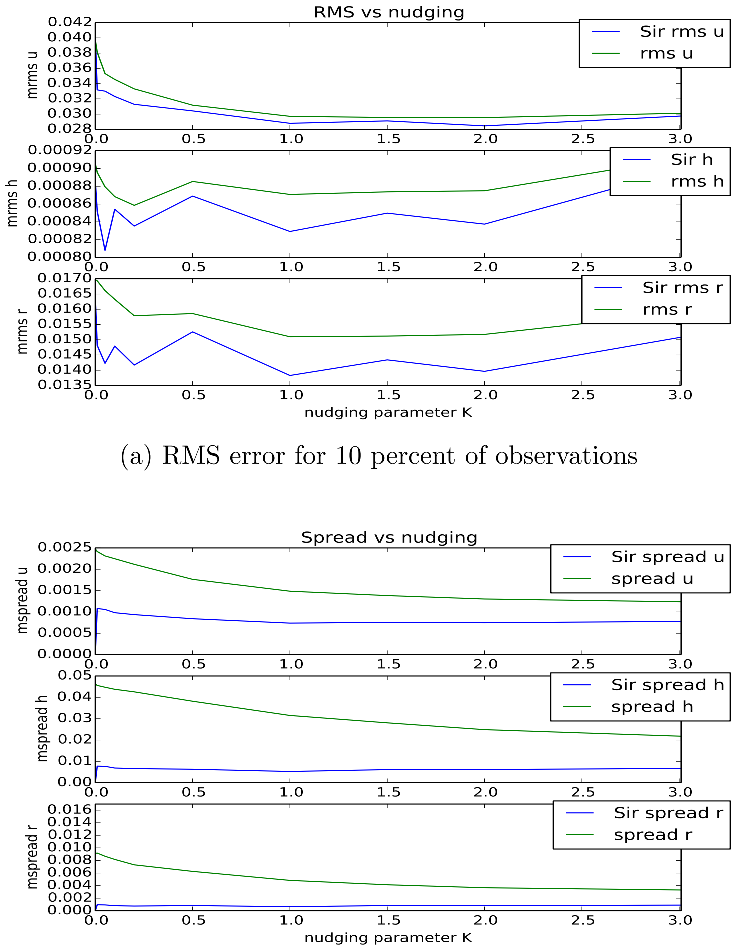


(a) RMS error for 40 percent of observations

(b) Spread for 40 percent of observations

Figure 5.24: Assimilation of 2 variables (wind and rain), RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of two variables, for observations at 40 percent of the domain and for a non-diagonal **K**-matrix.

Coverage of 10% of observations for two assimilated variables



(a) RMS error for 10 percent of observations

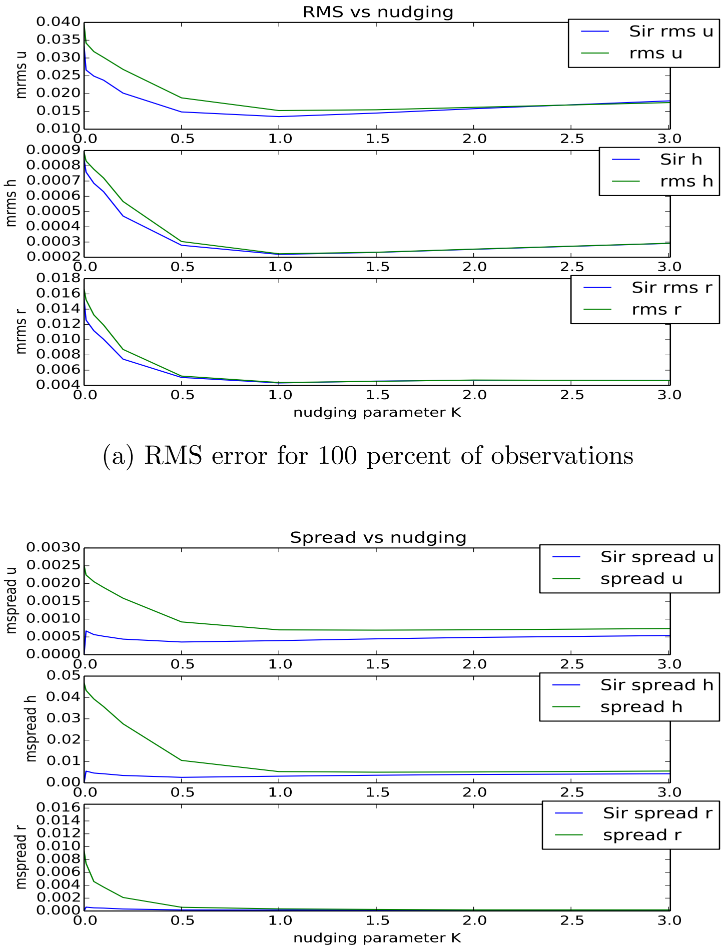
(b) Spread for 10 percent of observations

Figure 5.25: Assimilation of 2 variables (wind and rain), RMS error (a) and spread (b) per nudging parameter K, of all variables, for the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), with assimilation of two variables, for observations at 10 percent of the domain and for a non-diagonal $\mathbf{K}$-matrix.

As before, the minimal RMS error increases with decreasing number of observations and of assimilated variables.

Figures 5.26-5.28 show results of the same spatial decrease in observations, from 100 over 40 to 10 percent, for the assimilation of wind velocity only. As in the previous experiments, the reduction of observations from 40 to 10 percent leads to a shift of the absolute minimum of the RMS error, especially of water height and rain, towards smaller nudging parameters (after a slight shift to higher nudging values when passing from 100 to 40 percent).

Coverage of 100% of observations for one assimilated variables

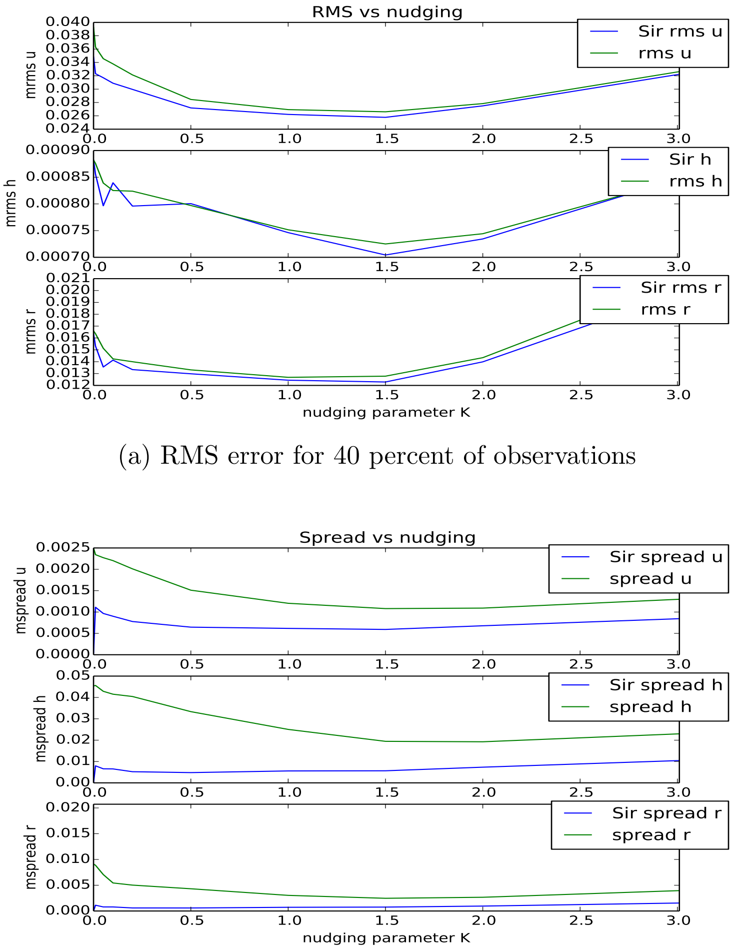


(a) RMS error for 100 percent of observations

(b) Spread for 100 percent of observations

Figure 5.26: Assimilation of 1 variable (wind), RMS (a) and spread (b) per nudging parameter K for the assimilation of wind velocity $\mathbf{u}$ with the efficient particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), for observations at every grid point and for a non-diagonal $\mathbf{K}$-matrix.

Coverage of 40% of observations for one assimilated variables

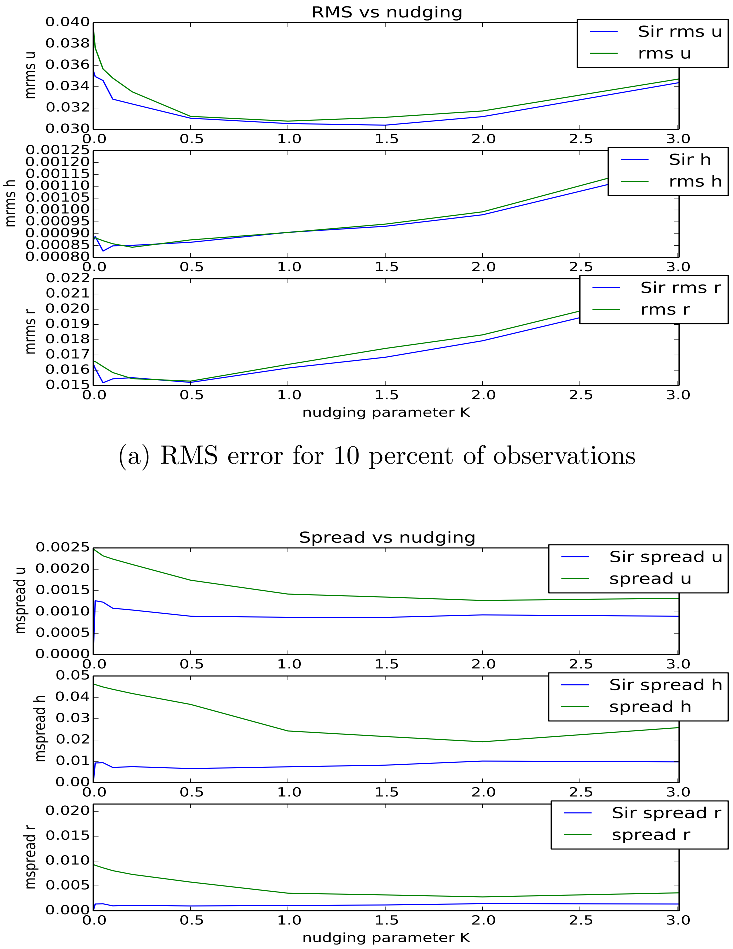


(a) RMS error for 40 percent of observations

(b) Spread for 40 percent of observations

Figure 5.27: Assimilation of 1 variable (wind), RMS error (a) and spread (b) per nudging parameter K for the assimilation of wind velocity $\mathbf{u}$ with the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), for observations at 40 percent of the domain and for a non-diagonal $\mathbf{K}$-matrix.

Coverage of 10% of observations for one assimilated variables



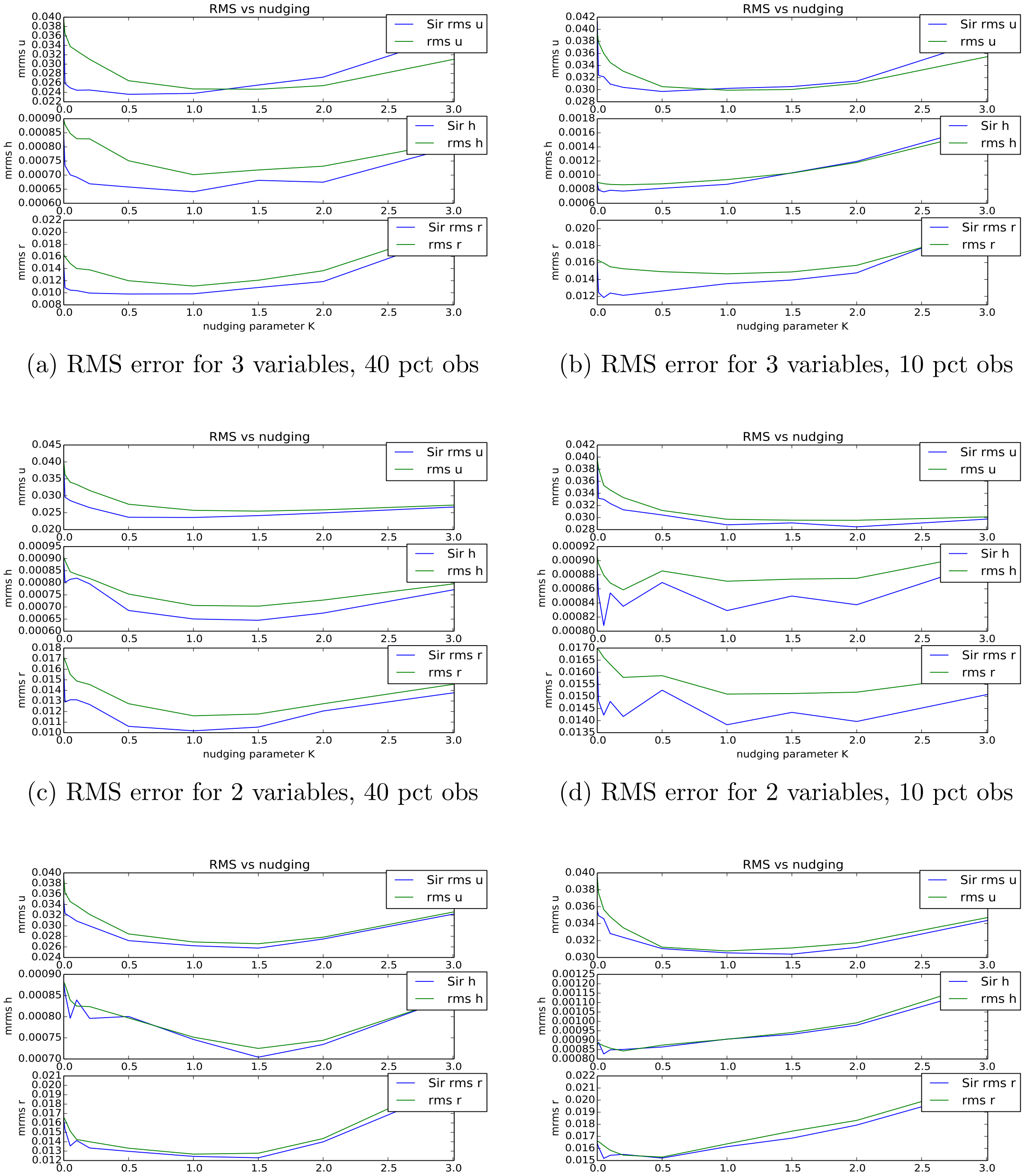
(a) RMS error for 10 percent of observations

(b) Spread for 10 percent of observations

Figure 5.28: Assimilation of 1 variable (wind), RMS error (a) and spread (b) per nudging parameter K for the assimlation of wind velocity $\mathbf{u}$ with the particle filter (blue line, denoted by 'SIR') vs a nudged ensemble without resampling (green line), for observations at 10 percent of the domain and for a non-diagonal $\mathbf{K}$-matrix.

Conclusive comparisons

Figure 5.29 shows the comparative RMS errors of the efficient particle filter using a non-diagonal $\mathbf{K}$-matrix, between 40 percent (left column of figures) and 10 percent (right column) of observations among the grid points. And finally in Figures 5.30-5.32, we compare the errors of the filter using a non-diagonal $\mathbf{K}$-matrix (left column of figures) and a diagonal $\mathbf{K}$-matrix. This is done for the assimilation of 3 (Fig. 5.30), 2 (Fig. 5.31) and 1 (Fig. 5.32) variables, respectively.



(a) RMS error for 3 variables, 40 pct obs

(b) RMS error for 3 variables, 10 pct obs

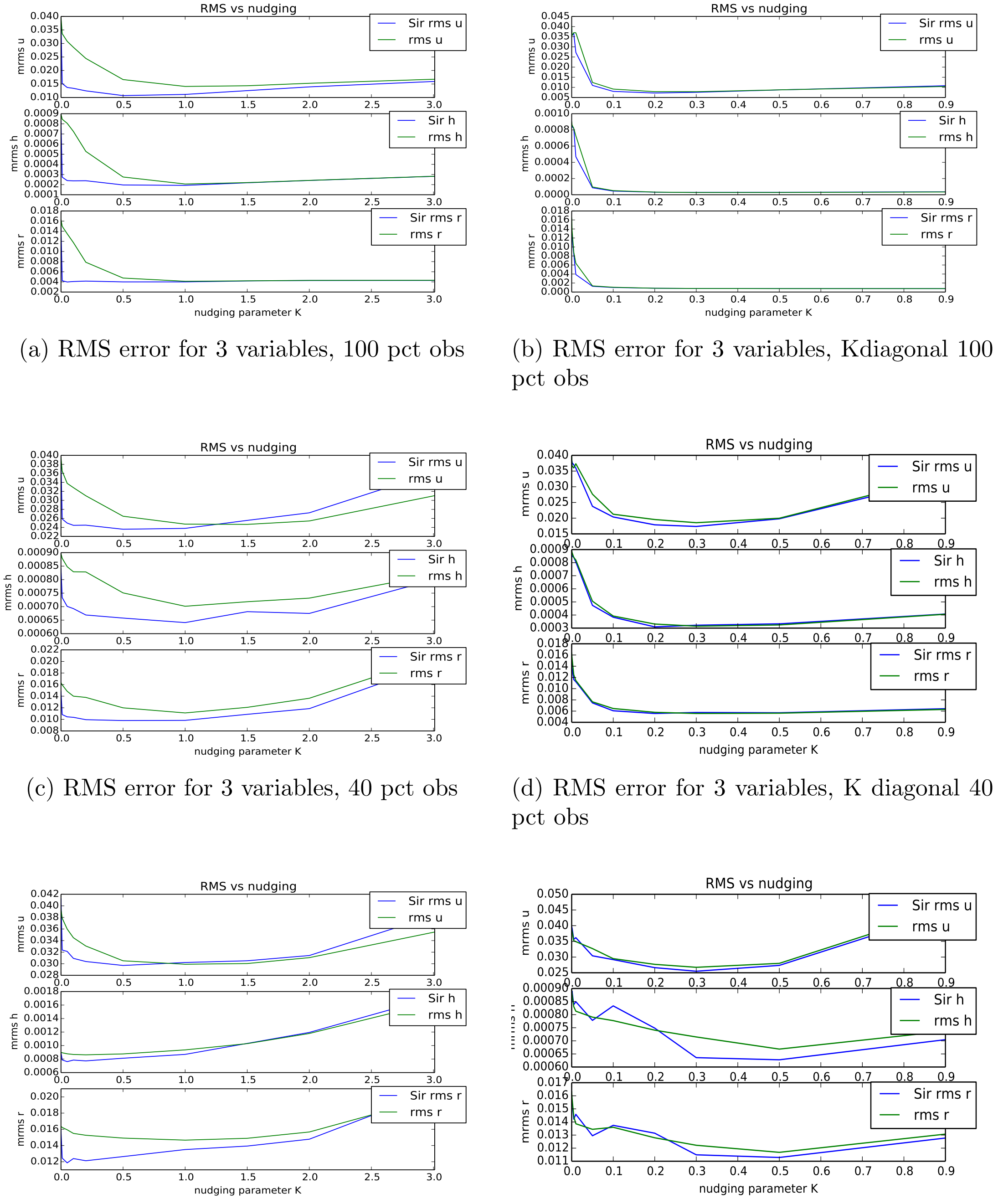
(c) RMS error for 2 variables, 40 pct obs

(d) RMS error for 2 variables, 10 pct obs

(e) RMS error for 1 variable, 40 pct obs

(f) RMS error for 1 variable, 40 pct obs

Figure 5.29: Comparison of the RMS errors, for an ensemble assimilated with the efficient particle filter vs a nudged-only ensemble, for a non-diagonal **K**-matrix, between 40 percent of available observations (left column of figures) and 10 percent of observations (right column of figures). The more variables are assimilated, the smaller the minimal error of all variables becomes. The percentage of observations is abbreviated by 'pct obs'.



(a) RMS error for 3 variables, 100 pct obs

(b) RMS error for 3 variables, Kdiagonal 100 pct obs

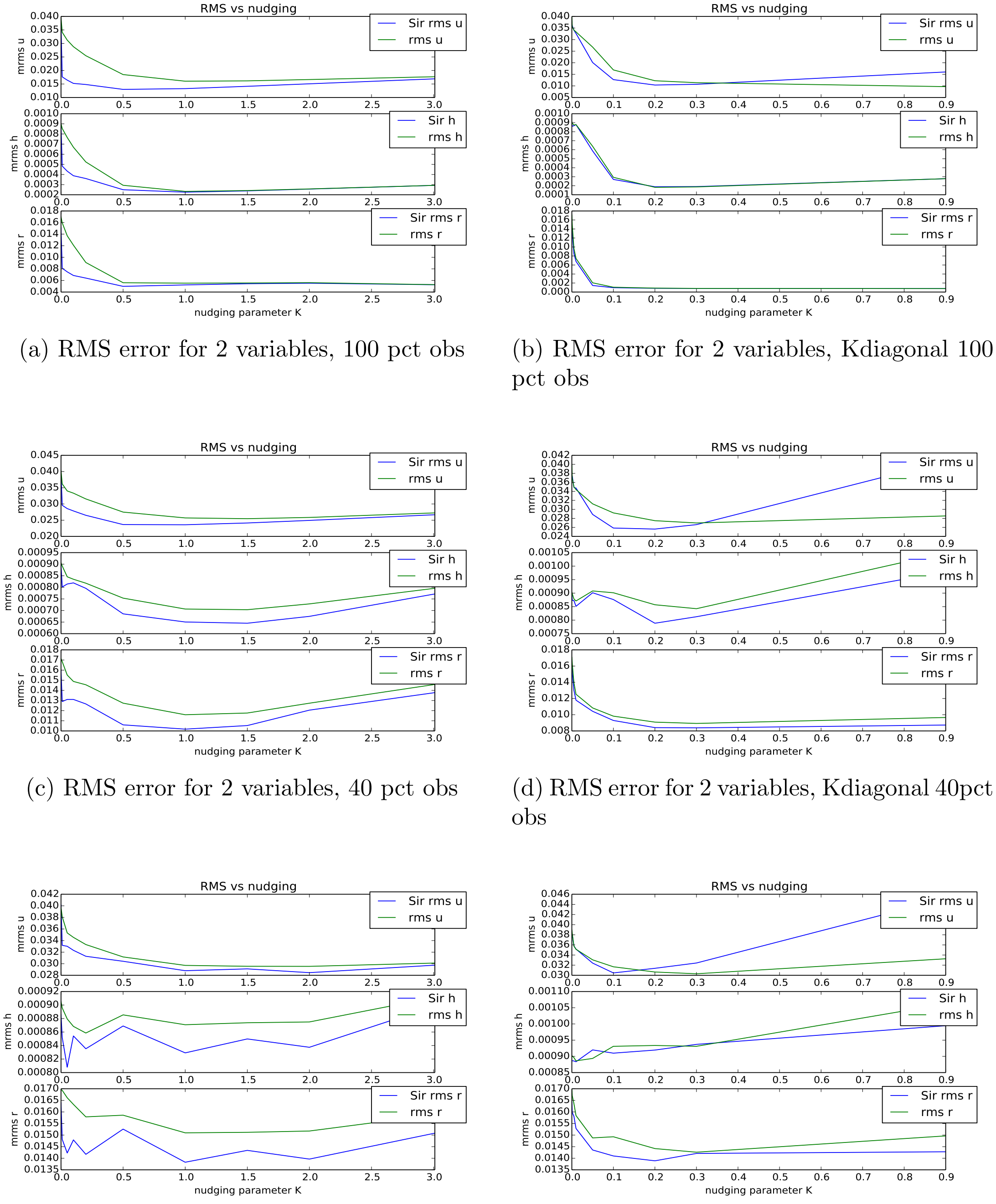
(c) RMS error for 3 variables, 40 pct obs

(d) RMS error for 3 variables, K diagonal 40 pct obs

(e) RMS error for 3 variables, 10 pct obs

(f) RMS error for 3 variables, Kdiagonal 10 pct obs

Figure 5.30: Comparison of the RMS per nudging parameter for the assimilation of 3 variables (**u**, **h**, **r**) with a non-diagonal **K**-matrix (left column of figures) vs a diagonal **K**-matrix (right column). The percentage of observations is abbreviated by 'pct obs'.



(a) RMS error for 2 variables, 100 pct obs

(b) RMS error for 2 variables, Kdiagonal 100 pct obs

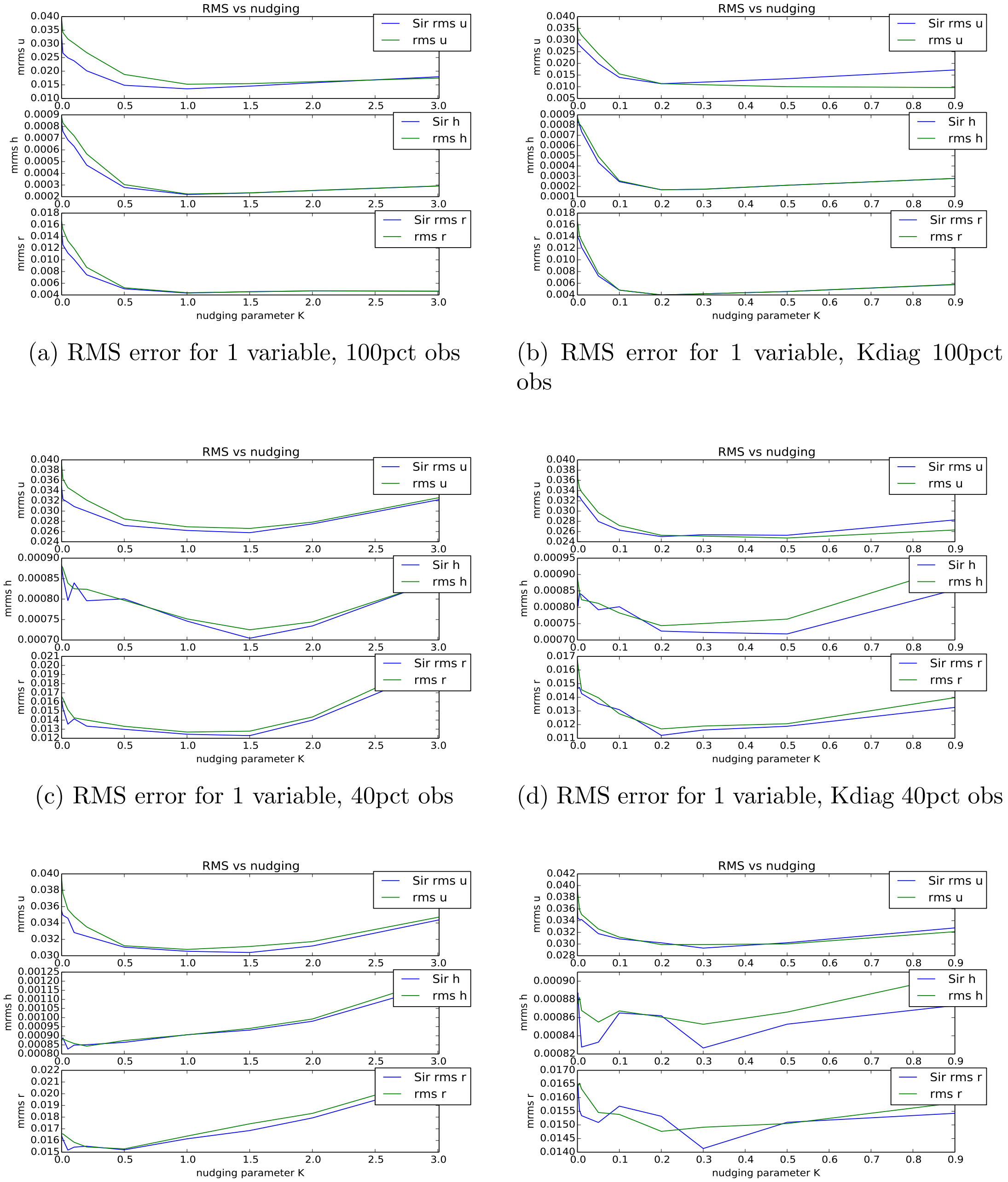
(c) RMS error for 2 variables, 40 pct obs

(d) RMS error for 2 variables, Kdiagonal 40pct obs

(e) RMS error for 2 variables, 10 pct obs

(f) RMS error for 2 variables, Kdiagonal 10 pct obs

Figure 5.31: Comparison of the RMS per nudging parameter for the assimilation of 2 variables (**u**, **r**) with a non-diagonal **K**-matrix (left column of figures) vs a diagonal **K**-matrix (right column). The percentage of observations is abbreviated by 'pct obs'.



(a) RMS error for 1 variable, 100pct obs

(b) RMS error for 1 variable, Kdiag 100pct obs

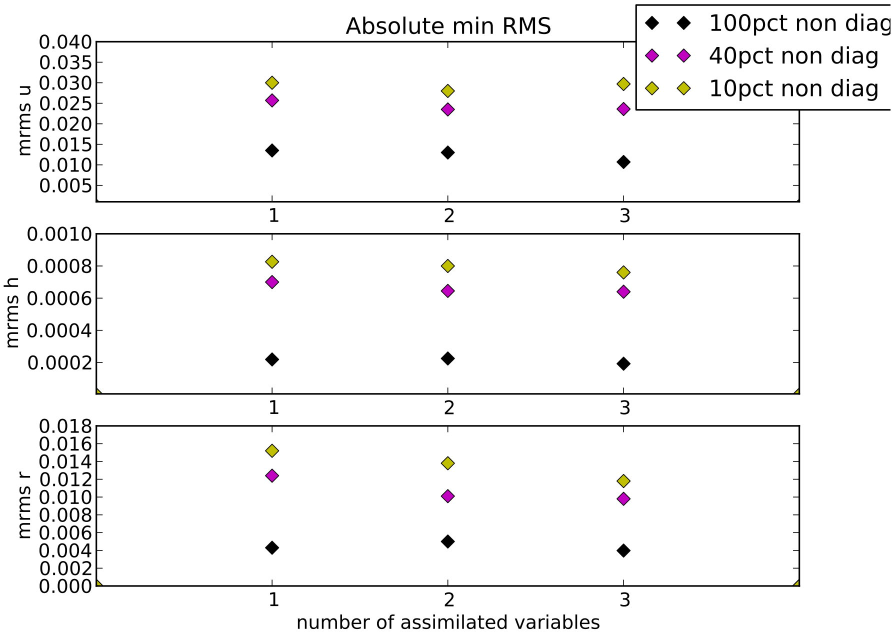
(c) RMS error for 1 variable, 40pct obs

(d) RMS error for 1 variable, Kdiag 40pct obs

(e) RMS error for 1 variable, 10pct obs

(f) RMS error for 1 variable, Kdiag 10pct obs

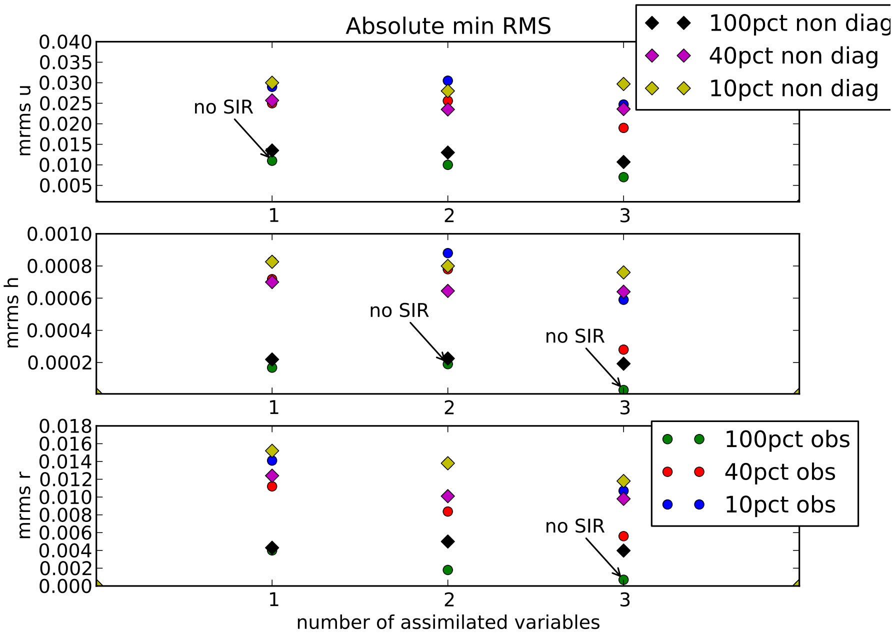
Figure 5.32: Comparison of the RMS per nudging parameter for the assimilation of 1 variable (**u**) with a non-diagonal **K**-matrix (left column of figures) vs a diagonal **K**-matrix (right column). The percentage of observations is abbreviated by 'pct obs'.



(a) Assimilation for a non-diagonal **K**-matrix

Figure 5.33: Absolute minimal mean RMS error for **u** (top panel), **h**, and **r** (bottom panel) per number of assimilated variables. The mean is taken over the mean RMS error obtained from the last 70 timesteps, taken over 5 realizations of the experiment.

Figure 5.33 shows, similarly to Figure 5.19, the resulting absolute minimal mean RMS errors of **u**, **h** and **r** as a function of the number of assimilated variables, for the case of applying the efficient particle filter using a non-diagonal nudging matrix **K**. Contrarily to the case with a diagonal **K**-matrix, the application of a non-diagonal nudging matrix seems to guarantee a systematically better performance of the efficient particle filter compared to nudging only. Figure 5.33 also indicates a reduction of the RMS error, in all variables, when assimilating more variables and when increasing the number of observations. The seemingly smaller RMS error of wind velocity in the assimilation of 2 variables and 10 percent observation coverage compared to 3 variables is probably due to stochastic fluctuations.



(a) Comparison of minimal RMS errors between use of diagonal and of non-diagonal **K**-matrix

Figure 5.34: Total comparison of minimal RMS errors between assimilation with a diagonal, and a non-diagonal **K**-matrix. Absolute minimal mean RMS error for **u** (top panel), **h**, and **r** (bottom panel) per number of assimilated variables. The mean is taken over the mean RMS error obtained from the last 70 timesteps, taken over 5 realizations of the experiment.

5.3.4 Conclusion

Based on our choice of model and observation error perturbation covariance matrices **Q** and **R** respectively, we observe a notable improvement with the efficient particle for the assimilation of all three variables when using a diagonal nudging matrix rather than a non-diagonal one. Best results in terms of minimal mean RMS error are obtained in the unrealistic case of observations being available everywhere, followed by the case of 40 percent and finally 10 percent of observation coverage. As expected, the RMS error decreases with an increased amount of assimilated observations or of dynamical variables. The improvement due to the choice of a diagonal **K**-matrix is, roughly speaking, of the order of 20 percent for **u**, of 50 percent for **h** and of 40 percent for **r** for the assimilation of three variables. When assimilating one variable (wind), there is no substantial difference between using a diagonal or a non-diagonal nudging matrix. When two variables (wind and rain) are assimilated, a non-negligible benefit of using a diagonal rather than a non-diagonal **K**-matrix can be observed for rain and water height, while it is negligible for wind. In the case of 100% observation coverage for a diagonal nudging matrix, nudging only performs as well as the particle filter.

To conclude, the best possible results in terms of minimum mean RMS errors are achieved for the unrealistic case where observations are available at every grid point, but with a nudging-only regime. For any kind of reduced information (in terms of density of observations in space, or of number of assimilated variables), the resulting error of the filter increases. But in this constellation of reduced information only, the efficient particle filter does notably better than only nudged ensembles of particles without resampling.

5.4 Applicability conditions for the filter

The results of the efficient particle filter applied on the modified shallow model in Section (5.3) reveal that when all three variables are assimilated, the use of a *diagonal nudging matrix* $\mathbf{K}$ instead of a $\mathbf{K}$-matrix of the form $\mathbf{K} = K\mathbf{Q}\mathbf{H}^T\mathbf{R}^{-1}$ as in [1], where we took a $\mathbf{Q}$-matrix as in (5.4), leads to lower RMS errors and hence to better results. However, depending on the nudging parameter, the efficient filter with a diagonal nudging matrix *does not necessarily perform better* than when particles are *only nudged*, without being replaced by other, presumably 'better placed' ones in the frame of a resampling procedure.

Since the filter seems to be sensitive to parameters such as nudging intensity, the covariances of model and observation errors, we therefore want to investigate further what drives its selection mechanism. The questions we address in this sections are:
1.) Under which conditions does resampling outperform mere nudging?
2.) If resampling outperforms nudging, what is leading to this result?

In Section (5.4.1), we analyze the behaviour of the density exponents involved in the resampling mechanism. We show that there is a quantity that provides information about the likelihood of the resampling procedure to select the right particles that will contribute to reducing the RMS error. This quantity is the standard deviation of the different density terms in the calculation of the weights. It is therefore interesting to derive an analytic estimate of this quantity. This is what we do in Section (5.4.2), where we obtain an order-of-magnitude estimate of the standard deviation. The theoretical estimate for the standard deviations is derived for the case where particles are not equalized. These derivations demonstrate the existence of an upper limit on the nudging parameter K. The particle filter cannot be expected to perform well if the nudging parameter exceeds this limit.

5.4.1 A complex selection mechanism

Resampling rewards comparatively high weights by replicating their corresponding particle. The weights are given by

$$\begin{aligned} w_i \sim exp\big(-\frac{1}{2} &\underbrace{(\mathbf{d}^n - \mathbf{H}\mathbf{x}^n)^T \mathbf{R}^{-1} (\mathbf{d}^n - \mathbf{H}\mathbf{x}^n)}_{1} - \\ -\frac{1}{2} &\underbrace{\sum_{j=n-L+1}^{n} (\mathbf{K}(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}^{j-1})) + \hat{\boldsymbol{\beta}}^j)^T \mathbf{Q}^{-1} (K(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}^{j-1})) + \hat{\boldsymbol{\beta}}^j)}_{2} \\ +\frac{1}{2} &\underbrace{\sum_{j=n-L+1}^{n} (\hat{\boldsymbol{\beta}}^j)^T \hat{\mathbf{Q}}^{-1} (\hat{\boldsymbol{\beta}})^j}_{3} \big) \end{aligned} \tag{5.21}$$

where, according to our test setting, $\hat{\mathbf{Q}} = \mathbf{Q}$, $\mathbf{K} = K\mathbf{Id}$ $(k \in [0, 1])$ is the diagonal nudging matrix and where $\mathbf{H}$ is the (projection) observation operator. We denote the exponent terms of the densities by 1, 2 and 3 for the considerations below, and define the standard deviation as in (5.17), replacing the state vector for each particle by the value of the density exponent in terms 1, 2, or 3.

Among the mechanisms which influence the weights' magnitude, there is the present distance separating observation from model *at assimilation time*, determined by the observation density. It is on this *final* distance (term 1 in equation (5.21)) that is based the computation of the RMS error, and a reduction of the latter which will be achieved by reducing the former. The other influence on the weights comes from nudging, reflected by a weighted sum of distances between observation and past states (term 2 in (5.21)). In the absence of nudging, the particle filter is identical to the SIR filter. Increasing nudging will lead to a reduced distance between former model states, within an assimilation cycle, and present observation, and hence to a smaller negative contribution in the weight exponents in term 2. On the other hand, this negative contribution to the weight exponent is increased by a squared nudging term which penalizes the weight. The more a particle is nudged, the smaller is its probability to reach this state because its model error is increased by the amount of nudging, and the smaller becomes its weight.

One question is what mechanism leads to a selection of particles such that their RMS error is being reduced ? It turns out that what drives the selection mechanism is not the absolute mean value, among particles, of the weight exponent (since the mean is discarded from the weights by normalization), but the variability of the exponent terms among the ensemble members. Hence, the standard deviation of the different parts 1, 2 and 3 constituting the exponents of the weights deserve special attention, since they are suspected to play a crucial role in the particle selection that will render resampling efficient.

On the other hand, what is the limit of nudging? At what point are created erroneous model imbalances by perturbing too much the variable fields so that the model prohibits nudging, independently of resampling? Such imbalances themselves drive particles away from the observation, thus increasing the distances between model and observation at each integration time step so that even nudging cannot reduce the negative weight contributions, leading to very low weights and a failure of the selection mechanism. Hence, depending on the model's reaction on relaxation, we expect a limit of nudging being responsible for the resampling mechanism to function above some nudging magnitude.

We try to answer these questions in what follows by analyzing the behaviour of the standard deviation of different parts of the weight exponent given by the observation density, and the p- and q-densities. For these numerical tests, we take into account the equalization of weights that charcaterizes the efficient particle filter according to [1] and [33].

Selection mechanism for the assimilation of one variable

Table (5.2) shows the mean standard deviations over time of the different density exponents 1, 2, 3 (defined in (5.21)) as a function of the increasing nudging intensity K.

K	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(1) - \sigma(2-3)$	benefit of SIR (in percent)
0.001	225.87	279.12	197.62	28.24	27.37
0.01	164.39	242.16	191.12	14.22	9.85
0.05	134.10	228.85	193.37	16.26	3.08
0.08	180.29	246.85	191.61	24.78	15.89
0.1	162.73	236.27	188.93	22.97	18.97
0.3	88.14	212.83	192.67	-0.27	-2.37
0.4	105.69	228.83	191.42	-10.94	-13.56
0.8	200.70	344.57	191.89	-84.72	-59.81

Table 5.2: Numerical standard deviations of exponent terms 1, 2 and 3 in the weights and corresponding percentual benefit of the efficient particle filter with weight equalization compared to a nudged-only ensemble of particles, depending on the nudging intensity. With increasing nudging intensity paramater K, the benefit of the filter is reduced as $\sigma(2-3)$ becomes dominant over $\sigma(1)$, until the nudged-only regime outperforms the filter with resampling. These tests were made during the assimilation of wind velocity for observations available at every grid point, and the nudging matrix used is a diagonal.

The standard deviations, with respect to the particles, of these terms were computed at each assimilation cycle before being averaged over the total number of 100 cycles. Term 2-3 corresponds to the exponent of the density fraction p/q. The last numerical values on the right indicate the benefit, in percent of the total mean RMS error (mean taken over the last 70 percent of assimilation time steps), of the efficient filter compared to a regime with nudging the particles only.

With increasing nudging intensity (denoted by K), the benefit of resampling becomes smaller and smaller until it is reversed and the ensemble of nudged-only particles gives a lower RMS error than the efficient particle filter. In such a situation the filter can be expected to fail because the selection mechanism does not replicate particles better placed with respect to the observation anymore.

From an intermediate nudging of $K \sim 0.3$ on, *when the selection mechanism* in the particle filter *starts giving worse results* than if particles were not resampled, we notice that the standard deviation $\sigma(1)$ of the observational term becomes *smaller* than the standard deviation of the term (2-3). In this situation, instead of being selected via the dominant variability of the distance between particles and observation, the particles are selected on the basis of term $\sigma(2-3)$, now dominant over $\sigma(1)$. In other words, the smallest value that the standard deviation $\sigma(2-3)$ can take and that leads to the largest weight, decides upon a particle being selected or not. It can also be observed that the relative improvement of the efficient filter with respect to the nudged regime is the best for very weak nudging (27 percent for $K = 0.001$).

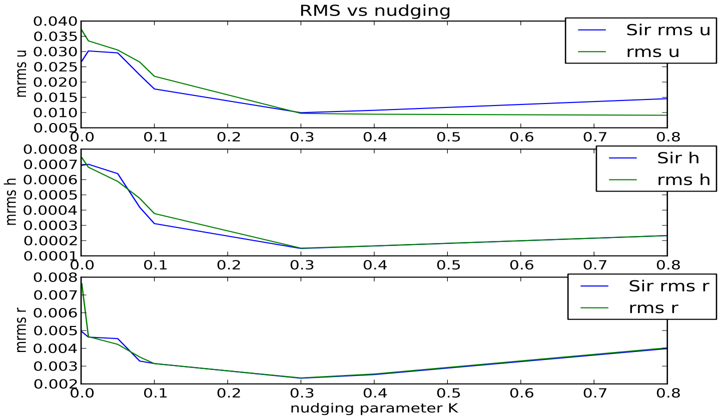


Figure 5.35: RMS error per nudging intensity parameter K, of wind velocity (upper panel), water height and rain (bottom panel), for the assimilation of wind velocity, for observations available at every grid point and a diagonal nudging matrix. The efficient particle filter with weight equalization (blue line) partly outperforms the nudged-only ensemble (green line) of particles until a nudging parameter of $K \sim 0.3$. Above this nudging limit, the efficient filter cannot not surpass a nudging-alone regime anymore in terms of RMS error. These standard deviation values represent only the wind velocity.

Figure 5.35 illustrates the mean RMS error per nudging parameter of all variables **u**, **h** and **r**, for the efficient particle filter compared to a nudged-only ensemble of particles, obtained for the above experiment. For this purpose, the individual RMS errors were computed at each assimilation time step for a fixed nudging intensity, before taking the mean of these errors over the last 70 assimilation time steps (out of 100 assimilation time steps). Figure 5.35 shows that the minimal mean RMS error achieved by the filter does not coincide with the best relative improvement, in terms of percentage, of the filter compared to the nudged-only ensemble.

K	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(1) - \sigma(2-3)$	benefit of SIR (in percent)
0.00	51.66	0.0	0.0	51.66	5.73
0.01	115.42	214.48	190.28	17.22	7.83
0.05	98.41	219.04	194.58	13.00	5.71
0.1	81.36	212.50	198.98	10.60	1.25
0.2	87.93	210.00	197.26	13.95	-1.12
0.5	76.63	204.55	190.58	8.21	-1.43
0.8	130.30	243.90	192.00	-4.60	-0.25

Table 5.3: Standard deviations of terms 1, 2 and 3 per increasing nudging parameter K for the assimilation of one variable (wind velocity) and 10 percent of spatially available observations, using a diagonal nudging matrix. When $\sigma(1)$ is dominated by $\sigma(2-3)$ for stronger nudging, the particle filter with resampling is tendentially outperformed by a nudging-only regime of particles without resampling. The values represent the wind velocity only.

Table 5.3 contains the numerical values of standard deviations of the density exponents for $\mathbf{u}$ per increasing nudging parameter K for the assimilation of one variable and 10 percent of spatially available observations, while Figure 5.36 illustrates the corresponding RMS error per nudging parameter for the efficient filter vs a nudged-only ensemble. These values are also the total density exponents, since terms from other variables vanish due to $\hat{\mathbf{Q}} = \mathbf{Q}$ and to $\mathbf{K} = K\mathbf{H}^T$ in the weight exponents. Beyond a medium nudging of $K \sim 0.5$ and for stronger nudging, resampling acts to the detriment to the efficient particle filter, as well for the assimilated variable $\mathbf{u}$ as for the rain $\mathbf{r}$, while only nudged particles produce a smaller RMS error. At the same time, the standard deviation of the observational term 1 becomes smaller in magnitude than term 2-3, providing observations less importance in the resampling process as the nudging parameter is increased (cf. Table 5.3).

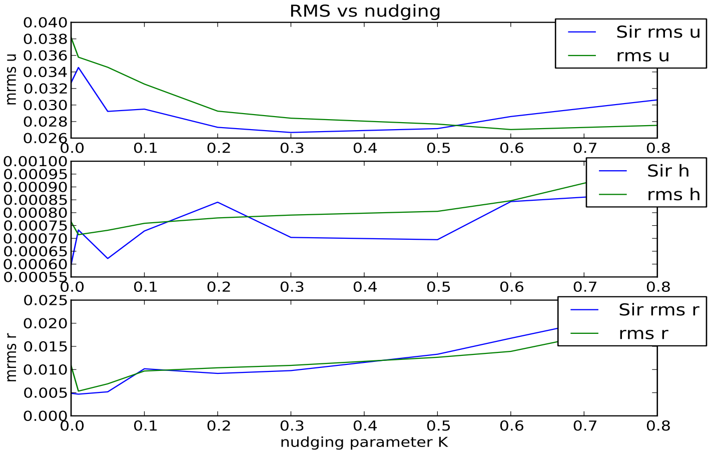


Figure 5.36: RMS error per nudging parameter for the efficient filter (blue line) vs a nudged-only ensemble (green line). Only wind velocity is assimilated, for observations available at 40 percent of the domain, and for a diagonal nudging matrix.

Selection mechanism for the assimilation of two variables

When assimilating two variables (here wind $\mathbf{u}$ and rain $\mathbf{r}$), the selection mechanism is somewhat different. Figure 5.4 shows the evolution, with increasing nudging intensity K, of the standard deviation of terms 1, 2 and $2-3$ for wind and rain separately (for the case where observations are available at 40 percent of the grid points). While in the wind variable, resampling leads to a medium improvement of maximally 12 percent (for $K = 0.1$), it is *largely beneficial to the rain* variable for any nudging parameter. The highest percentage of improvement of the filter with respect to a nudged-only ensemble is attained with a peak of 67 for $K = 0.3$.

There are no contributions in the standard deviations coming from water height since, again due to the choice of $\hat{\mathbf{Q}} = \mathbf{Q}$ and $\mathbf{K} = K\mathbf{H}^T$, the stochastic terms involving $\mathbf{h}$ in (5.21) cancel out and since water height is not observed. In this situation where two variables are contributing to the weight formation, the *standard deviation of the p-density exponent* (and hence term $(2-3)_r$) in the rain variable is *clearly dominating* over all other terms, having values of the order of 10^8. Hence, it is the past evolution of the rain contained in the p-density which plays a dominant role in the selection of particles. Those particles with the smallest negative p-density exponent will be selected first.

On top of this 'highest ranking' selection mechanism (ruled by the p-density of $\mathbf{r}$) comes the role of the observation density. The nudging parameter corresponding to the *smallest standard deviation* $\sigma_{\mathbf{r}}(1)$ of the rain observation, $K = 0.3$ (with a standard deviation of $\sigma_{\mathbf{r}}(1) = 31.38$), is *rewarded by the highest improvement* percentage (67 percent) of the filter in the rain variable. We presume that the particle whose rain-component has the smallest distance to observation at the last time step, on top of being the 'winning' variable (compared to wind) in the *a priori* selection process through term 2, will have the highest chance to be reselected. However, the term 1 variablity should not be too small (below a value of 10 according to our tests) in order to keep an impact on the selection mechanism.

K	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(1) - \sigma(2-3)$	benefit of SIR (in percent)
u					
0.00	60.92	0.0	0.0	60.92	12.97
0.01	187.57	203.35	198.75	148.45	3.76
0.05	195.67	1224.94	197.40	-1004.78	5.75
0.1	238.81	4897.17	195.50	-4651.01	12.34
0.2	375.42	16451.97	194.44	-1678.92	9.85
0.5	1014.76	61967.84	192.63	-60950	-15.52
0.8	5545.23	36962.36	196.61	-358416.24	-99.39
r					
0.0	24.14	0.0	0.0	24.14	76.01
0.01	78.55	514066.86	194.19	-513988.56	26.95
0.05	65.40	12.86e+6	198.92	-12.86e+6	28.20
0.1	80.57	51.98e+6	199.44	-51.98e+6	40.59
0.2	47.30	99.38e+6	199.57	-99.38e+6	64.29
0.5	46.94	190.53e+6	196.40	- 190.53e+6	59.66
0.8	117.98	274.44e+6	196.42	- 274.44e+6	44.39

Table 5.4: Standard deviation of terms 1,2, and 3 with the corresponding improvement of the efficient filter compared to a nudging-only regime of particles, for the assimilation of two variables (wind velocity and rain), 40 percent of observations and using a diagonal nudging matrix. The values in the top of the figure represent the wind, while the values in the bottom of the figure represent rain.

For the wind variable, we notice that resampling leads to a worse result after the maximum benefit percentage in rain is obtained, for $K = 0.3$. Beyond this nudging intensity, there is a strong discontinuity in the wind's standard deviation of term 1 which jumps from an order of 100 to an order of 1000.

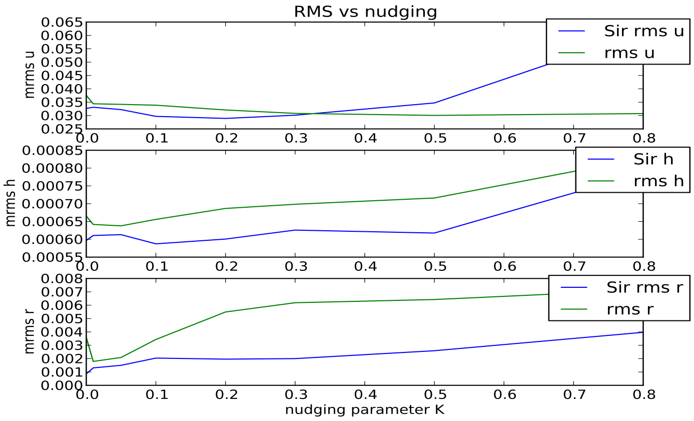


Figure 5.37: RMS error per nudging parameter for the efficient particle filter (blue line) vs a nudged-only ensemble (green line) for the assimilation of wind and rain. The graph corresponds to the numerical values in Table 5.4.

Furthermore, the maximal percentage of improvement of the RMS error with respect to a nudging-alone regime roughly corresponds to the minimum RMS error in the wind that can be achieved by the filter.

All these numerical values are random and subject to strong variability. But there are some visible tendencies. For both variables **u** and **r** separately, smaller standard deviations of term 1 ensure smaller negative exponent values, hence larger weights and a higher benefit of resampling. When the standard deviation of the observation term increases too much with intensified nudging, the percentage of improvement due to resampling diminishes.

However, it can be noticed that rain has an 'advantage' over wind which is visible throughout all assimilation tests (cf. Figures 5.30-5.32). While in the wind variable, there is a threshold of nudging, above which the nudged-only ensemble performs significantly better than the efficient filter with resampling, it is not true for the rain. In the rain variable, the filter does never worse than a nudged-only ensemble. But we emphasize that this is only the case for the current choice of observation and model error parameters (q and r). For other choices of parameters, the situation can be different (for example when the observation variance in rain is larger). Table 5.4 shows that term 1 deviations in **r** are smaller than in **u** and the values do not increase as fast with increased nudging than wind. Hence, rain is more 'nudging-resisitent' than wind, although the RMS error starts to increase again after a nudging parameter where the minimum of term 1 was attained. It is thus to be expected that rain will have larger benefits with resampling than the wind.

Figure 5.37 shows the large improvement of the efficient filter for the rain variable over the whole range of nudging parameters, unlike for the wind variable. This suggests that the rain variable is not very correlated to the wind variable, contrarily to the water height, which also benefits from the resampling as does the rain.

K	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(1) - \sigma(2-3)$	benefit of SIR (in percent)
u					
0.00	46.51	0.0	0.0	46.51	12.98
0.01	133.20	189.92	189.57	107.50	2.74
0.05	88.16	351.73	206.96	-190.74	1.79
0.1	84.59	2198.98	207.11	-2080.14	9.52
0.2	233.35	7709.91	197.38	-7472.52	1.15
0.5	1238.40	39079.86	196.84	-37833.33	-22.64
0.8	3445.80	145857.39	197.68	-142411.88	-47.20
r					
0.0	0.04	0.0	0.0	0.04	49.25
0.01	41.30	26.79e+4	199.70	-26.78e+5	14.34
0.05	19.70	6.33e+6	196.31	-6.33e+6	30.59
0.1	35.42	2.62e+6	192.31	-2.62e+6	15.32
0.2	35.19	96.79e+6	203.49	-96.79e+6	47.50
0.5	94.13	227.10e+6	196.15	- 227.10e+6	41.23
0.8	120.87	259.61e+6	192.85	- 259.61e+6	16.76

Table 5.5: Standard deviation of terms 1, 2, and 3 with the corresponding improvement of the efficient filter compared to a nudging-only regime of particles, for the assimilation of two variables (wind velocity and rain), 10 percent of observations and using a diagonal nudging matrix.

A similar situation as in the example above can be observed for the case of 10 percent of available observations among the domain. Table 5.5 shows again the standard deviations of the different exponent terms of the densities involved in the computation of the weights, and Figure 5.38 its corresponding RMS error graph as a function of nudging. The largest contributions in terms of standard deviation come from term 2 in the rain variable. Therefore, the selection of particles will be based on the smallest accumulation, over all integration time steps since the last assimilation time, of p-density contributions, defined by distances between past state vectors and same future observation.

We again notice a rapid increase of the term 1 variability in the wind, which inflates dramatically beyond a nudging parameter of $K = 0.2$ and is accompanied by a worse performance of the filter compared to the nudged-only ensemble. At the same time, term 1 variability in rain is and remains smaller than in wind over the whole range of nudging parameters. The relative improvement of the filter is high until a strong nudging of $K = 0.8$.

The water height in Figure 5.38 shows the same behaviour as rain with resampling, a largely improved RMS error compared to the nudged-only ensemble over the whole range of nudging intensities. From former tests Figure 5.34, tough, we know that assimilating wind and rain does not improve the RMS error of the water height, compared to assimilating only wind. Only assimilating three variables reduces the error in **h** as well.

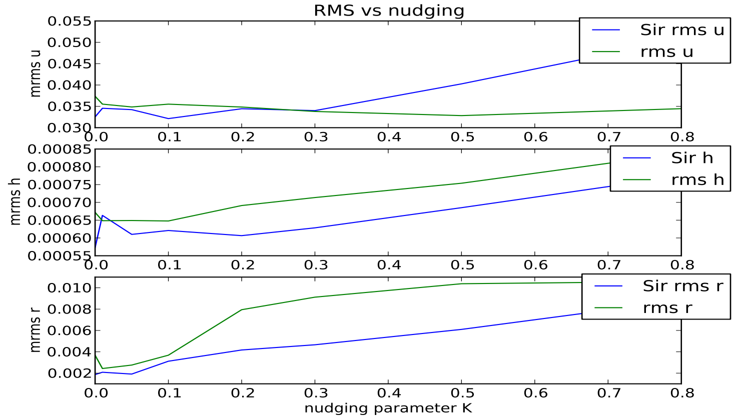


Figure 5.38: RMS error per nudging parameter for the efficient particle filter (blue line) vs a nudged-only ensemble (green line), with 10 percent of observations available on the domain. The graph made for assimilating two variables corresponds to the tests made for Figure 5.5.

Again and roughly speaking, the last time step in form of observation density influences the final selection of particles by favouring those particles which are closest, in their rain component, to the corresponding observation. Rain and water height are largely improved by resampling within the filter, while the wind RMS error is slightly improved for weak nudging but monotonically grows beyond medium nudging.

Selection mechanism for the assimilation of three variables

In the case of three variables being assimilated (cf. Table 5.6), the dominant terms are the two competing terms 2 in **h** and **r**. The selection of particles is expected to take place based on the smallest negative term 2 in water height and rain, and wind will not play an important part in the selection mechanism. The only significant improvement of wind with respect to a nudged-only ensemble is obtained for the very weakest nudging parameter $K = 0.01$. But since it cannot compete with other variables, this could be by pure chance.

K	$\sigma(1)$	$\sigma(2)$	$\sigma(3)$	$\sigma(2-3)$	$\sigma(1)-\sigma(2-3)$	benefit of SIR (in percent)
u						
0.00	78.16	0.0	0.0	78.16	6.14	
0.01	192.02	194.56	190.09	32.15	159.86	14.37
0.05	304.85	435.10	199.64	368.95	-82.099	4.38
0.1	465.32	1198.77	204.95	386.95	-699.82	0.76
0.2	1200.52	3419.70	196.27	3401.86	-2201.33	-1.31
0.5	8037.31	19617.55	192.13	19618.72	-11581.41	-13.21
0.8	16262.71	74893.65	188.52	74907.08	-58644.36	-25.59
h						
0.00	2.88	0.0	0.0	46.51	12.98	-3.63
0.01	7.72	4.21e+6	185.45	4.21e+6	-4.21e+6	15.03
0.05	13.32	29.63e+6	190.89	29.63e+6	-29.63e+6	8.61
0.1	15.29	54.91e+6	202.15	54.91e+6	-54.91e+6	-0.43
0.2	16.85	82.38e+6	201.40	82.38e+6	-82.38e+6	-1.15
0.5	15.86	12.68e+7	192.88	121.74e+6	-12.68e+7	3.27
0.8	7.41	16.46e+7	190.36	164.61e+6	-16.46e+7	4.70
r						
0.0	1.33	0.0	0.0	0.0	1.33	44.62
0.01	766.76	5.02e+6	186.75	5.02e+6	-5.02e+6	53.57
0.05	263.39	23.48e+6	194.12	23.48e+6	-23.48e+6	25.26
0.1	116.93	48.74e+6	199.49	48.74e+6	-48.74e+6	37.88
0.2	34.79	55.42e+6	186.97	55.42e+6	-55.42e+6	-7.47
0.5	7.69	66.62e+6	188.93	66.62e+6	-66.62e+6	-10.67
0.8	3.68	63.96e+6	196.71	63.96e+6	-63.96e+6	4.17

Table 5.6: Standard deviation of terms 1,2, and 3 with the corresponding improvement of the efficient filter compared to a nudging-only regime of particles, for the assimilation of three variables (wind velocity, water height and rain), 40 percent of observations and using a diagonal nudging matrix.

The **h**-variable becomes dominant in its term 2 for stronger nudging than $K = 0.05$, to the detriment of **r**. Once the variable **h** becomes the leading variable in the selection, and since it is then 'failing' for increasing nudging parameters, the other variables are not expected to lead to a benefit of the filter with respect to nudging-only. We notice a special correlation with **r**. It shows a similar behaviour as **h**, although it is not dominant in the selection of particles.

We further notice that the observational term in the water height is very low compared to the other variables, and that the sudden discontinuous decrease of term 1 variability in rain beyond $K = 0.1$ coincides with the latter's worse performance of resampling. This raises the question whether too weak a standard deviation of term 1 is detrimental to the filter.

Figure 5.39 shows the corresponding graph of the RMS error as a function of the nudging parameter. Beyond a nudging parameter of $K = 0.1$, the filter fails in all variables with respect to a nudging-only regime, and the rain variable is the last one to keep a positive impact of resampling.

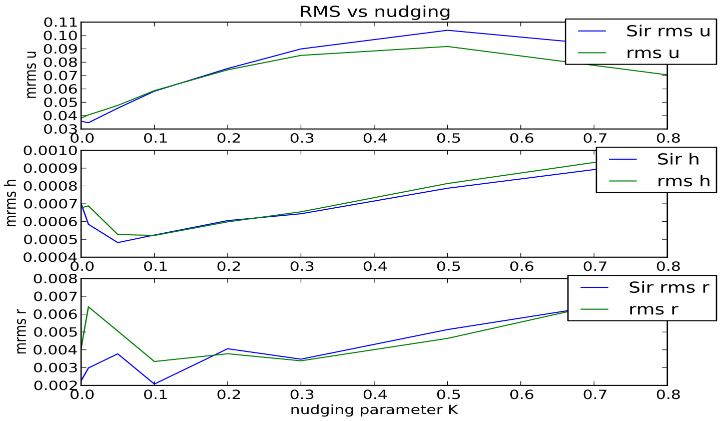


Figure 5.39: RMS error per nudging parameter for the efficient particle filter (blue line) vs a nudged-only ensemble (green line), with 40 percent of observations available on the domain. The graph made for assimilating three variables corresponds to the tests made for Figure (5.6).

Remark

In conclusion, whether a particle filter with resampling is likely to surpass a nudged ensemble depends on a certain number of parameters, at first the model and observation error covariances $(\mathbf{Q}, \mathbf{R})$ as well as the nudging parameter K. These are responsible, in addition to the deterministic physical model which drives the evolution of the particles, for the magnitude of the variability of terms 1, 2 or 3 in the weight exponents that is given to the different variables. It is a complex mechanism of interaction of all these parameters that constitute the basis of construction of the weights, and one should take into account that these ratios of magnitudes between different density exponents may be responsible of some ordering, in terms of 'priority', of the selection behaviour. In the above tests, we showed results only using a diagonal nudging matrix since it potentially leads to lower RMS errors (cf. Section 5.3.3).

The question that arises following the fact that the magnitude of a density exponent rules the selection of particles invites to try to formulate a 'criterium of success' or at least some kind of analytical answer to the question why resampling works, and when ? For a given set of parameters (error covariances, domain size, observation size, ...) that we have in this setup, what is the limit in nudging that we can apply before failure of the efficient filter takes place ? Is there such a limit ? In order to answer these questions, we derive an estimate of the standard deviation for the different exponent terms in the next Section.

5.4.2 Variance of the densities

Let us consider for now the case where the given observations are available at every grid point, i.e. where $\mathbf{H} = \mathbf{Id}$. Then, (5.21) becomes

$$\begin{aligned} w_i \sim exp\big(&- \frac{1}{2} \underbrace{(\mathbf{d}^n - \mathbf{x}^n)^T \mathbf{R}^{-1} (\mathbf{d}^n - \mathbf{x}^n)}_{1} - \\ &\underbrace{-\frac{1}{2} \sum_{j=n-L+1}^{n} (\mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{j-1})) + \hat{\boldsymbol{\beta}}^n)^T \mathbf{Q}^{-1} (\mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{j-1})) + \hat{\boldsymbol{\beta}}^n)}_{2} \\ &+ \frac{1}{2} \underbrace{\sum_{j=n-L+1}^{n} \hat{\boldsymbol{\beta}}^T \mathbf{Q}^{-1} \hat{\boldsymbol{\beta}}}_{3} \big). \end{aligned} \tag{5.22}$$

We derive a theoretical estimate of the standard deviation for each of the density terms 1,2,3 introduced in the above expression. In the following we will use the Einstein convention for summing over double indices. The variance of a random variable $\mathbf{y} \in \mathbb{R}^n$ is defined as

$$\begin{aligned} \mathrm{Var}(\mathbf{y}) &= E[(\mathbf{y} - E[\mathbf{y}])^2] \\ &= E[\mathbf{y}^2] - E[\mathbf{y}]^2. \end{aligned} \tag{5.23}$$

We will compute the variance of the expression 1 and 2-3 above by substituting them into formula (5.23). In what follows, we will denote the nudging parameter by k instead of K, for reasons of notational ease.

Variance of the observation-density term 1

We replace $\mathbf{x}^n$ by the expression $\mathbf{x}^n = f(\mathbf{x}^{n-1}) + \mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{n-1})) + \boldsymbol{\beta}^n$ of the equation that governs the generated state vector, in the exponent of the observation density in (5.21). We have

$$\begin{aligned} 1^2 =& [(\mathbf{d}^n - \mathbf{x}^n)^T \mathbf{R}^{-1} (\mathbf{d}^n - \mathbf{x}^n)]^2 = \\ &= [(\mathbf{d}^n - f(\mathbf{x}^{n-1}) - \mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{n-1})) - \boldsymbol{\beta}^n)^T \mathbf{R}^{-1} \\ &\times (\mathbf{d}^n - f(\mathbf{x}^{n-1}) - \mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{n-1})) - \boldsymbol{\beta}^n)]^2 \\ &= [(\mathbf{b} - \boldsymbol{\beta}^n)^T \mathbf{R}^{-1} ((\mathbf{b} - \boldsymbol{\beta}^n)]^2 \end{aligned} \tag{5.24}$$

where

$$\mathbf{b} := (\mathbf{Id} - \mathbf{K})(\mathbf{d}^n - f(\mathbf{x}^{n-1})) \tag{5.25}$$

is a deterministic term in the above expression. After expanding the quadratic terms in (5.24) and taking advantage of the fact that $\mathbf{R}$ is diagonal, we have

$$\begin{aligned} E(1^2) &= [(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}) + (\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n - 2(\mathbf{b}^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)]^2 \\ &= E[(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b})^2] + E[\big((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n\big)^2] + \\ &+ 4E[\big((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\mathbf{b}\big)^2] + \\ &+ 2E[(\mathbf{b}^T R^{-1}\mathbf{b})((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)]. \end{aligned} \tag{5.26}$$

Other terms vanish, since uneven powers of $\boldsymbol{\beta}^n$ have mean zero. This can be seen from the following consideration. The terms in (5.26) represent moments of the random Gaussian vector $\boldsymbol{\beta}^n$ up to the fourth order. There exists a one-to-one correspondance between the generating function g of a random vector $\boldsymbol{\beta}^n$ and the distribution of this vector [1]. The Taylor development of g is [2]

$$g_{\boldsymbol{\beta}^n}(\mathbf{k}) = E[e^{\boldsymbol{\beta}^n\mathbf{k}}] \approx 1 + \frac{\mathbf{k}^T\mathbf{Q}\mathbf{k}}{2} + \frac{1}{2!}\left(\frac{\mathbf{k}^T\mathbf{Q}\mathbf{k}}{2}\right)^2, \quad \mathbf{k} \in \mathbb{R}^N. \tag{5.27}$$

Therefore, we have - leaving out the upper time index n for the moment - the identities

$$E[\boldsymbol{\beta}] = \frac{\partial g(\mathbf{k})}{\partial \mathbf{k}}, \quad E[\boldsymbol{\beta}^2] = \frac{\partial^2 g(\mathbf{k})}{\partial \mathbf{k}^2}, \quad E[\boldsymbol{\beta}^3] = \frac{\partial^3 g(\mathbf{k})}{\partial \mathbf{k}^3}, \quad \text{... etc.} \tag{5.28}$$

From (5.27), it is obvious that, like in the scalar case, uneven powers of $\boldsymbol{\beta}$ have zero mean.

Let us now approximate the terms in (5.26).

- The first term in (5.26) is deterministic and hence equal to

$$E[(\mathbf{b^TR^{-1}b)^2}] = (\mathbf{b^TR^{-1}b)^2}. \tag{5.29}$$

A further approximation can be made for $\mathbf{b}$ in (5.25). Using the fact that $\mathbf{K}$ and $\mathbf{Id}$ are diagonal and introducing a parameter, $(d-f)$ such that

$$(\mathbf{d}^n - f(\mathbf{x}_{n-1})) \approx (d-f)\mathbf{e}_N \quad \mathbf{e}_N = (1, ..., 1)^T, \tag{5.30}$$

where $\approx$ means a rough approximate given that we suppose that all components of the vector $\mathbf{d}^n - f(\mathbf{x}_{n-1})$ have the same value, we obtain

$$\mathbf{b} = (\mathbf{Id} - \mathbf{K})(\mathbf{d}^n - f(\mathbf{x}^{n-1})) \approx (1-k)(d-f)\mathbf{e}_N, \tag{5.31}$$

which is a very rough estimation. It follows for a diagonal $\mathbf{R}$-matrix

$$E[(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b})^2] = (\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b})^2 \approx ((1-k)^2(d-f)^2r^{-1}N)^2, \tag{5.32}$$

where N is the dimension of a state vector.

[1] see for ex. [19], ch. 9

[2] cf. online course of Robert G. Gallagher, http://www.rle.mit.edu/rgallager/PDFS/Gauss.pdf, Ex. 2.6. p.83

- The second term in (5.26) corresponds to the fourth-power moment of $\boldsymbol{\beta}$. We have

$$E[((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)^2] = E[((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)] = r^{-2}E[\boldsymbol{\beta}^4], \tag{5.33}$$

since $\mathbf{R} := r\mathbf{Id}$ is diagonal. $E[\boldsymbol{\beta}^4]$ can be calculated as the fourth-order term of the Taylor development in (5.27)

$$\begin{aligned}
E[\boldsymbol{\beta}^4] &= \left.\frac{\partial^4 g(\mathbf{k})}{\partial k_n \partial k_l \partial k_j \partial k_i}\right|_{\mathbf{k}=\mathbf{0}} \\
&= \frac{1}{8}\frac{\partial^4}{\partial k_n \partial k_l \partial k_j \partial k_i}\left((k_l Q_{ln} k_n)(k_i Q_{ij} k_j)\right) \\
&= \frac{1}{4}\frac{\partial^3}{\partial k_n \partial k_l \partial k_j}\left[\left(\delta_{il} Q_{ln} k_n + k_l Q_{ln} \delta_{in}\right) k_i Q_{ij} k_j\right] \\
&= \frac{1}{4}\frac{\partial^2}{\partial k_n \partial k_l}\left(\frac{\partial}{\partial k_j}\left[(Q_{in} k_n + k_l Q_{li}) k_l Q_{ln} k_n\right]\right) \\
&= \dots \\
&= Q_{lj}Q_{in} + Q_{jn}Q_{il} + Q_{ij}Q_{ln} \\
&= \delta_{ln}\delta_{ji}[2tr(\mathbf{Q}^2) + Q_{ij}Q_{ln}] \\
&= 2tr(\mathbf{Q}^2) + N^2q^2,
\end{aligned} \tag{5.34}$$

where N is the dimension of $\mathbf{Q}$ and q is its (in our case) diagonal element. Due to the specific design of the Q-matrix, which has two off-diagonals representing a correlation over four gridpoints, we proceed to a rough estimation

$$2tr(\mathbf{Q}^2) \approx \frac{13}{4}Nq^2, \tag{5.35}$$

which is confirmed by numerical computation.

Proof. Indeed, let us write $\mathbf{Q}^2$. From

$$\mathbf{Q}^2 = \begin{pmatrix}
q & \frac{q}{2} & \frac{q}{4} & 0 & \dots & \dots & 0 \\
\frac{q}{2} & q & \frac{q}{2} & \ddots & 0 & \dots & 0 \\
\frac{q}{4} & \frac{q}{2} & q & \ddots & & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \frac{q}{4} \\
\vdots & \ddots & \ddots & \ddots & \ddots & q & \frac{q}{2} \\
0 & \dots & \dots & 0 & \frac{q}{4} & \frac{q}{2} & q
\end{pmatrix}
\begin{pmatrix}
q & \frac{q}{2} & \frac{q}{4} & 0 & \dots & \dots & 0 \\
\frac{q}{2} & q & \frac{q}{2} & \ddots & 0 & \dots & 0 \\
\frac{q}{4} & \frac{q}{2} & q & \ddots & & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \frac{q}{4} \\
\vdots & \ddots & \ddots & \ddots & \ddots & q & \frac{q}{2} \\
0 & \dots & \dots & 0 & \frac{q}{4} & \frac{q}{2} & q
\end{pmatrix}, \tag{5.36}$$

it follows that

$$
\begin{aligned}
tr(\mathbf{Q}^2) = & \\
=& tr\begin{pmatrix}
q^2+(\frac{q}{2})^2+(\frac{q}{4})^2 & \cdots & \cdots & \cdots \\
\vdots & (\frac{q}{2})^2+q^2+(\frac{q}{2})^2+(\frac{q}{4})^2 & \cdots & \\
\vdots & \ddots & (\frac{q}{4})^2+(\frac{q}{2})^2+q^2+(\frac{q}{2})^2+(\frac{q}{4})^2 & \cdots \\
\vdots & \ddots & \ddots & \cdots \\
 & \cdots & & \cdots
\end{pmatrix} \\
=& Nq^2 + 2(N-2)(\frac{q}{2})^2 + 2(N-4)(\frac{q}{4})^2 + 2(\frac{q}{2})^2 + 4((\frac{q}{4})^2) \\
=& Nq^2 + 2(N-1)(\frac{q}{2})^2 + 2(N-2)(\frac{q}{4})^2 \\
=& [N + \frac{N-1}{2} + \frac{N-2}{8}]q^2 \\
=& [N\frac{13}{8} - \frac{3}{4}]q^2. \qquad (5.37)
\end{aligned}
$$

For large N, this number can be approximated as

$$tr(\mathbf{Q}^2) \approx \frac{13}{8}Nq^2, \qquad (5.38)$$

which is confirmed by comparing a numerical value of $8.11e-12$ for $N = 500$ and $q = 1.0e-7$. □

Hence, the second term in (5.26) becomes

$$E[((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)^2] = r^{-2}E[\boldsymbol{\beta}^4] \approx \frac{13}{4}Nq^2r^{-2} + N^2q^2r^{-2}, \qquad (5.39)$$

- The mixed, third term in (5.26), becomes

$$
\begin{aligned}
E[\big((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\mathbf{b}\big)^2] &= E[\big((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\mathbf{b}\big)\ \big((\boldsymbol{\beta}^n)^T R^{-1}\mathbf{b}\big)] \\
&= r^{-2}E[\big((\boldsymbol{\beta}^n)^T\mathbf{b}\big)\ \big((\boldsymbol{\beta}^n)^T\mathbf{b}\big)] \\
&= r^{-2}b_iE[\boldsymbol{\beta}_i^n\boldsymbol{\beta}_j^n]b_j \qquad (5.40) \\
&= r^{-2}b_iQ_{ij}b_j \\
&= (1-k)^2r^{-2}(d-f)^2\sum_{ij}Q_{ij}.
\end{aligned}
$$

$\sum_{i,j} Q_{ij}$ in turn can be well approximated by

$$\sum_{i,j} Q_{ij} \approx \frac{5}{2}qN, \qquad (5.41)$$

which again is confirmed by the numerical verification $\sum_{ij} Q_{ij} \approx 2.48Nq$.

Proof. For, indeed, summing up the elements of $\mathbf{Q}$ (represented in (5.36)) leads to

$$\begin{aligned}\sum_{i,j} Q_{ij} &= Nq + 2(N-1)\frac{q}{2} + 2(N-2)\frac{q}{4} \\ &= [\frac{5}{2}N - 2]q, \qquad (5.42)\end{aligned}$$

which as in the previous proof, turns out to be

$$\sum_{i,j} Q_{ij} \approx \frac{5}{2}Nq \qquad (5.43)$$

for large N. □

Hence, the third term in (5.26) can be approximated as

$$E[\big((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\mathbf{b}\big)^2] \approx \frac{5}{2}(1-k)^2r^{-2}(d-f)^2qN. \qquad (5.44)$$

- The last term in (5.26) can be roughly estimated in a similar way. As before, and in virtue of the fact that $\boldsymbol{\beta}^n$ is Gaussian,

$$\begin{aligned}E[(\mathbf{b^T}R^{-1}\mathbf{b})((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)] &= (\mathbf{b^TR^{-1}b})\mathbf{E}[((\beta^\mathbf{n})^\mathbf{T}\mathbf{R^{-1}}\beta^\mathbf{n})] \\ &\approx (1-k)^2(d-f)^2r^{-1}NE[((\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n)] \\ &= (1-k)^2(d-f)^2r^{-2}NE[(\boldsymbol{\beta}^n)^2] \\ &= (1-k)^2(d-f)^2r^{-2}NQ_{ii} \\ &= (1-k)^2(d-f)^2r^{-2}N^2q. \qquad (5.45)\end{aligned}$$

With all the approximated terms, we finally get

$$\begin{aligned}E[1^2] \approx &((1-k)^2(d-f)^2r^{-1}N)^2 + \frac{13}{4}Nq^2r^{-2} + r^{-2}N^2q^2 + \\ &+10(1-k)^2r^{-2}(d-f)^2qN + \\ &+2(1-k)^2(d-f)^2N^2r^{-2}q. \qquad (5.46)\end{aligned}$$

After having estimated $E[1^2]$, we can now estimate the term $E^2[1]$. Using the same notations and simplifications as before, we separate again the deterministic term $\mathbf{b}$ and the stochastic term $\boldsymbol{\beta}$ in the expression, and get

$$\begin{aligned}E^2[1] =&E^2[(\mathbf{d}^n - \mathbf{x}^n)^T\mathbf{R}^{-1}(\mathbf{d}^n - \mathbf{x}^n)] \\ =&E^2[(\mathbf{d}^n - f(\mathbf{x}^{n-1}) - \mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{n-1})) - \boldsymbol{\beta}^n)^T\mathbf{R}^{-1} \\ &\times (\mathbf{d}^n - f(\mathbf{x}^{n-1}) - \mathbf{K}(\mathbf{d}^n - f(\mathbf{x}^{n-1})) - \boldsymbol{\beta}^n)] \\ =&E^2[(\mathbf{b} - \boldsymbol{\beta}^n)^T\mathbf{R}^{-1}((\mathbf{b} - \boldsymbol{\beta}^n)] \\ =&\big(E[(\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}) - 2(\mathbf{b}^T)\mathbf{R}^{-1}\boldsymbol{\beta}^n + (\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n]\big)^2 \\ =&\big((\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}) + E[(\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}\boldsymbol{\beta}^n]\big)^2. \qquad (5.47)\end{aligned}$$

Single power terms in $\boldsymbol{\beta}$ vanish because the mean of $\boldsymbol{\beta}$ is zero, by assumption. Then, using (5.32) for the deterministic part and the computation steps used in (5.45), we find

$$\begin{aligned} E^2[1] &\approx \big((1-k)^2(d-f)^2r^{-1}N + r^{-1}Nq\big)^2 \\ &= ((1-k)^2(d-f)^2r^{-1}N)^2 + 2(1-k)^2(d-f)^2N^2r^{-2}q + r^{-2}N^2q^2. \end{aligned} \tag{5.48}$$

It follows from 5.46 and 5.48 that the variance of the first term in the exponent of the weights in (5.21) is

$$\begin{aligned} \mathrm{Var}(1) &= E[(1-E[1])^2] \\ &= E[1^2] - E^2[1] \\ &= \frac{13}{4}Nq^2r^{-2} + 10Nqr^{-2}(1-k)^2(d-f)^2. \end{aligned} \tag{5.49}$$

Variance of the (p-q)-density term 2-3

We derive similarly an estimate for the standard deviation of the term 2-3 in (5.21). Using the abbreviation $\boldsymbol{\gamma}^j := \mathbf{d}^n - f(\mathbf{x}^{j-1})$, we expand the square term $(2-3)^2$,

$$E[(2-3)^2] = \sum_{j=n-L+1}^{n} E[\big((\mathbf{K}\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j + 2(\mathbf{K}\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^j\big)^2] \tag{5.50}$$

$$= \sum_{j,l}^{n} \big((\mathbf{K}\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j\big)\big((\mathbf{K}\boldsymbol{\gamma}^l)^T\mathbf{Q}^{-1}K\boldsymbol{\gamma}^l\big) + \underbrace{4\sum_{j,l}^{n} E[(\mathbf{K}\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^j(\mathbf{K}\boldsymbol{\gamma}^l)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^l]}_{\text{(i)}},$$

where we left out, as before, linear terms in $\boldsymbol{\beta}$ which by definition do not give a contribution to the mean and where we took acount of the linearity of the latter. Furthermore, using again the fact that $\mathbf{K}$ is diagonal and $\mathbf{Q}$ symmetric and extracting the deterministic terms out of the expectation sign,

$$\begin{aligned} (i) &= \sum_{j,l}^{n} E[(\mathbf{K}\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^j(\mathbf{K}\boldsymbol{\gamma}^l)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^l] \\ &= k^2\sum_{j,l}^{n} \left(\gamma_i^j Q_{im}^{-1}\gamma_k^l\right) Q_{kn}^{-1}E[\beta_m^j\beta_n^l] \\ &= k^2\sum_{j,l}^{n} \left(\gamma_i^j Q_{im}^{-1}\gamma_k^l\right)\left(Q_{nk}^{-1}Q_{mn}\delta_{jl}\right) \\ &= k^2\sum_{l}^{n} \left(\gamma_i^l Q_{im}^{-1}\gamma_k^l\right)\left(Q_{nm}^{-1}Q_{mn}\delta_{km}\right) \\ &= k^2\sum_{l}^{n} \gamma_i^l Q_{im}^{-1}\gamma_m^l. \end{aligned} \tag{5.51}$$

We used the definition of the variance and $E[\beta_m^j \beta_n^l] = Q_{mn}\delta_{jl}$, where δ_{jl} is the delta function with respect to the time indices l and j. It follows that

$$E[(2-3)^2] \quad = \sum_{j,l}^{n} \left(((\mathbf{K}\boldsymbol{\gamma}^j)^T \mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j)(\mathbf{K}\boldsymbol{\gamma}^l)^T \mathbf{Q}^{-1} K \boldsymbol{\gamma}^l \right) + 4k^2 \sum_{l}^{n} \gamma_i^l Q_{im}^{-1} \gamma_m^l. \tag{5.52}$$

Since, taking again advantage of the fact that $E[\boldsymbol{\beta}^j] = \mathbf{0}$,

$$\begin{aligned} E^2[(2-3)] &= \left(\sum_{j=n-L+1}^{n} E[\left((\mathbf{K}\boldsymbol{\gamma}^j)^T \mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j + 2(\mathbf{K}\boldsymbol{\gamma}^j)^T \mathbf{Q}^{-1}\boldsymbol{\beta}^j\right)] \right)^2 \\ &= \left(\sum_{j=n-L+1}^{n} (\mathbf{K}\boldsymbol{\gamma}^j)^T \mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j \right)^2 \\ &= \sum_{j,l}^{n} \left(((\mathbf{K}\boldsymbol{\gamma}^j)^T \mathbf{Q}^{-1}\mathbf{K}\boldsymbol{\gamma}^j)(\mathbf{K}\boldsymbol{\gamma}^l)^T \mathbf{Q}^{-1} K \boldsymbol{\gamma}^l \right). \end{aligned} \tag{5.53}$$

After some straightforward cancellations of terms in (5.52) and (5.53) and using the same approximation (5.30) for $\boldsymbol{\gamma}^j = \mathbf{d}^n - f(\mathbf{x}^{j-1}) \approx (d-f)\mathbf{e}_N$, we finally can compute the variance of term (2-3),

$$\begin{aligned} \mathrm{Var}(2-3) &= E[(2-3)^2] - E^2[(2-3)] \\ &= 4k^2 \sum_{l}^{n} \left(\gamma_i^l Q_{im}^{-1} \gamma_m^l \right) \\ &\approx 4Lk^2 (d-f)^2 \sum_{im} Q_{im}^{-1} \\ &\approx \frac{8}{5} L k^2 N q^{-1} (d-f)^2. \end{aligned} \tag{5.54}$$

In the last step we used the approximation

$$\sum_{im} Q_{im}^{-1} \approx \frac{2}{5} N q^{-1}. \tag{5.55}$$

This value is in accord with the numerically computed value of $2.0e9$ for $N = 500$ and $q = 1.0e - 7$.

As a conclusion, we have the following rough estimates for the standard deviations of the observation and the p-q densities in the weights, respectively,

$$std(1) \approx \left(\frac{13}{4} N q^2 r^{-2} + 10 N q r^{-2} (1-k)^2 (d-f)^2 \right)^{\frac{1}{2}}, \tag{5.56}$$

$$std(2-3) \approx \left(\frac{8}{5} L k^2 N q^{-1} (d-f)^2 \right)^{\frac{1}{2}}. \tag{5.57}$$

Case of scarce observations ($\mathbf{H} \neq \mathbf{Id}$)

In the case of spatially scarce observations, the expression for the weight exponents in (5.21) contains the observation operator $\mathbf{H}$. The nudging matrix $\mathbf{K}$ and the deterministic vector $\mathbf{b}$ in (5.25) are replaced by

$$\mathbf{K} \in \mathbb{R}^{N\times N} \rightarrow \mathbf{KH}^T \in \mathbb{R}^{N\times m}, \tag{5.58}$$
$$\mathbf{b} \in \mathbb{R}^{N} \rightarrow \mathbf{b} := (\mathbf{Id}_m - \mathbf{HKH}^T)(\mathbf{d}^n - \mathbf{H}f(\mathbf{x}^{n-1})) \in \mathbb{R}^m, \tag{5.59}$$

respectively. From the observational term 1 in (5.24),

$$\begin{aligned} 1 =& [(\mathbf{d}^n - \mathbf{Hx}^n)^T\mathbf{R}^{-1}(\mathbf{d}^n - \mathbf{Hx}^n)]^2 \\ &= [(\mathbf{b} - \mathbf{H}\boldsymbol{\beta}^n)^T\mathbf{R}^{-1}((\mathbf{b} - \mathbf{H}\boldsymbol{\beta}^n)]^2 \end{aligned} \tag{5.60}$$

where $\mathbf{b}$ is given by (5.59), it follows that the estimate of term 1 in (5.56) is obtained by replacing N by m,

$$std(1) \approx \left(\frac{13}{4}mq^2r^{-2} + 10mqr^{-2}(1-k)^2(d-f)^2\right)^{\frac{1}{2}}. \tag{5.61}$$

This expression can be simplified to

$$std(1) \approx \left(10mqr^{-2}(1-k)^2(d-f)^2\right)^{\frac{1}{2}} \quad \text{if} \quad q^2r^{-2} \ll 1, \tag{5.62}$$

since the term $\frac{13}{4}mq^2r^{-2}$ in (5.61) can be neglected. Indeed, $q^2 \ll r^2$ is valid for our computations, since with our test settings we have $q_u = 10^{-7}$ and $r_u = 10^{-6}$. The resulting number of 10^{-2} is negligible compared to the numerical standard deviation of term 1, and to the right term in (5.61).

As to term $(2-3)$, now using $\boldsymbol{\gamma}^j := \mathbf{d}^n - \mathbf{H}f(\mathbf{x}^{j-1})$, its only remaining part (i) in (5.50), as before, now is

$$\begin{aligned} (i) &= \sum_{j,l=n-L+1}^{n} E[(\mathbf{KH}^T\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^j(\mathbf{KH}^T\boldsymbol{\gamma}^l)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^l] \\ &= k^2 \sum_{j,l=n-L+1}^{n} E[(\mathbf{H}^T\boldsymbol{\gamma}^j)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^j(\mathbf{H}^T\boldsymbol{\gamma}^l)^T\mathbf{Q}^{-1}\boldsymbol{\beta}^l] \\ &= k^2 \sum_{l=n-L+1}^{n} \gamma_i^l Q_{im}^{-1}\gamma_m^l, \end{aligned} \tag{5.63}$$

with the only difference that the vector $\mathbf{H}^T\boldsymbol{\gamma}^j \in \mathbb{R}^N$ has rank m instead of N and that therefore, only m^2 terms of the matrix $\mathbf{Q}$ are summed up. Returning to our old notation

K for the nudging parameter instead of k and taking into account the special shape of $\mathbf{Q}^{-1}$ and (5.55), we estimate term $(2-3)$ as

$$std(2-3) \approx \left(\frac{8}{5}LK^2mq^{-1}(d-f)^2\right)^{\frac{1}{2}}. \tag{5.64}$$

Finally, we have

$$std(1) \approx \left(10mqr^{-2}(1-K)^2(d-f)^2\right)^{\frac{1}{2}}, \quad q^2r^{-2} \ll 1 \tag{5.65}$$

$$std(2-3) \approx \left(\frac{8}{5}LK^2mq^{-1}(d-f)^2\right)^{\frac{1}{2}}. \tag{5.66}$$

5.4.3 Indication criterium for the success of resampling

The estimates (5.56) and (5.57) of the standard deviation of the different density exponents in the above section were derived in the absence of weight equalization [3]. The estimates can only be very rough, since the values encountered vary very much during the simulation. They are intended to give an *indication of the approximate order of magnitude* that the standard deviations are expected to take. For example, they help understand that the mean and the standard deviation of term 3 (derivation not shown here), i.e. of the exponent of the q-density, are always nearly constant and of the order of NL and $\sqrt{2NL}$ respectively, showing that this term is normalized and does not depend on the the magnitude of the model perturbation and that therefore, this can be taken into account in understanding which particles are chosen to be resampled.

We derived these values only for the case where just one variable, here the wind velocity, is assimilated. The comparison with numerical values is shown below (Table 5.7). The theoretical estimates were computed as solution, for the only unknown parameter $\lambda := \mathbf{d}^n - \mathbf{H}f(\mathbf{u}^{n-1})$, to the two equations (5.56) and (5.57). λ_1 refers to the parameter λ computed in equation (5.56), while λ_2 refers to the parameter derived from (5.57).

[3] We do not know how to estimate them, taking into account the last nearly deterministic step where the particles are nudged with a particular parameter α to confer them all the same target weight.

$m = N$	numerical $\sigma(1)$	theoretical $\sigma(1)$	num. $\sigma(2-3)$	theoret. $\sigma(2-3)$	λ_1^2/λ_2^2
K=0.1	69.11	83.77	369.8	336.19	0.8
K=0.2	33.88	39.55	391.9	340.6	0.74
K=0.3	19.2	27.0	459.5	341.29	0.54
K=0.4	15.05	21.88	577.59	414.10	0.50
K=0.5	11.43	18.38	730.01	466.90	0.40
K=0.6	8.48	14.78	878	514.4	0.33
K=0.7	6.27	11.52	1069.51	591.0	0.3
K=0.8	4.58	8.33	1324	740	0.3
K=0.9	3.22	4.62	1652.4	1170	0.49

Table 5.7: Numerical vs theoretical values of the standard deviations of the exponent terms 1 and 2-3 in the densities. The tests were made for a domain of $N = 200$ grid points, with observations at every grid point ($m = N$), 100 cycles and $L = 100$ integration time steps between two assimilation cycles, $q = 1.0e-7$, $r = 1.0e-6$, for an ensemble of $k = 40$ members.

$m \neq N$	numerical $\sigma(1)$	theoretical $\sigma(1)$	num. $\sigma(2-3)$	theoret. $\sigma(2-3)$	λ_1^2/λ_2^2
K=0.1	122.57	78.17	345.97	557.43	2.5666
K=0.2	76.91	36.53	363.41	789.72	4.6727
K=0.3	71.25	23.43	398.75	1271.23	9.8214
K=0.4	52.54	17.20	454.19	1463.38	10.1698
K=0.5	36.86	13.76	543.97	1548.98	0.5076
K=0.6	31.09	12.88	753.35	1949.55	6.3620
K=0.7	22.74	12.17	1108.57	2230.37	3.7993
K=0.8	10.49	8.66	1312.73	1809	1.7020
K=0.9	4.78	7.02	2327	1826	0.507

Table 5.8: Numerical vs theoretical values of the standard deviations of the exponent terms 1 and 2-3 in the densities. The tests were made for a domain of $N = 200$ grid points, $m = 25$ observations, 100 cycles and $L = 100$ integration time steps between two assimilation cycles, $q = 1.0e-7$, $r = 1.0e-6$, for an ensemble of $k = 40$ members.

Tables 5.7 and 5.8 show that the theoretical values of the standard deviations are *roughly in accord* with the numerical values, being of the same order of magnitude. In the case where one variable is assimilated, we emit the hypothesis (cf. 5.4.1) that the standard deviation of the observation density term 1 needs to surpass the standard deviation of the other density term $(2-3)$, in order to impose a selection of particles according to their last time step position with respect to observations. Hence, we can formulate such a criterium based on our theoretical results.

$$std(1)/std(2-3) \approx \frac{5}{2}\left(\frac{qr^{-2}(1-K)^2}{LK^2q^{-1}}\right)^{\frac{1}{2}}, \tag{5.67}$$

and therefore,

$$std(1) \gtrsim std(2-3) \Leftrightarrow \frac{1-K}{K} \gtrsim \frac{2r}{5q}\sqrt{L} \tag{5.68}$$

For the parameters used in testing the shallow water model, with $q = 1.0e-7$, $r = 1.0e-6$,$L = 100$, this leads to a nudging parameter which is very weak,

$$K \lesssim \frac{1}{\frac{2r}{5q}\sqrt{L}} \Leftrightarrow K \lesssim 2.5e-2. \tag{5.69}$$

This limit is only a rough indication of a limiting paramater k above which no 'successful' selection of particles can be expected anymore. It shows that the filter has a nudging limit which depends on the parameters involved in the application of the particle filter.

Effect of weight equalization

The above analytical limit (5.68), though, applies only to a regime without weight equalization procedure. It is for now not possible to derive an exact estimation of the standard deviations of the weight exponents when weight equalization is performed in the filter. In Figure 5.32 where weight equalization was applied, the limiting parameter for the success of the filter is $K \sim 0.2$, i.e about 10 times larger than our analytical estimate in (5.69). In order to investigate further what happens to the standard deviations under influence of weight equalization, we analyze the numerical values of terms $\sigma(1)$ and $\sigma(2-3)$ with and without weight equalization.

$m \neq N$	$\sigma(1)_{\mathrm{we}}$	$\sigma(1)$	$\sigma(2-3)_{\mathrm{we}}$	$\sigma(2-3)$	$\sigma(1)_{\mathrm{we}}/\sigma(1)$	$\frac{\sigma(2-3)_{\mathrm{we}}}{\sigma(2-3)}$	$\overline{\alpha}$
K=0.1	73.52	49.52	61.38	343.53	1.48	0.17	1.97
K=0.2	55.39	29.22	47.06	350.05	1.89	0.13	1.28
K=0.3	64.15	29.22	55.26	365.29	2.18	0.15	1.0
K=0.4	63.06	21.72	60.89	391.33	2.90	0.15	0.82
K=0.5	61.12	13.96	64.27	414.41	4.37	0.15	0.71
K=0.6	79.05	12.45	78.06	478	6.34	0.16	0.64
K=0.7	105.66	9.38	110.71	611.13	11.25	0.18	0.55
K=0.8	105.38	5.12	111.73	678.74	20.54	0.16	0.49
K=0.9	126.38	2.68	140.67	880.22	47.07	0.15	0.45

Table 5.9: Comparison between mean standard deviations of terms 1 and 2-3 in the densities for the case where weights are equalized vs the case where no weight equalization is performed. In addition, we print the mean (over ensemble members) parameter in the last time step nudging term α in $\alpha\mathbf{K}$. All standard deviations are averaged over 70 assimilation time steps (after excluding the first 10 ones). This test was made for $m = 25$ observations among $N = 200$ grid points, for 80 cycles and $L = 100$ integration time steps between two assimilation cycles, $q = 1.0e - 7$, $r = 1.0e - 6$, for an ensemble of $k = 40$ members.

Table 5.9 shows the numerical values of the standard deviations of the density exponents 1 and $2-3$ for the case where equalization of weights is being performed, compared to the case without equalization at the last integration time step. In the right colums are printed the fractions of $\sigma(1)$ with weight equalization and $\sigma(1)$ without weight equalization. The same is done for $\sigma(2-3)$. It turns out that the standard deviation of observational term 1 becomes larger in the frame of weight equalization, while the standard deviation of term $2-3$ decreased significantly. Here, we see that the result is that up to nudging $K = 0.5$, the standard deviation of term 1 dominates though not significantly over term $2-3$ (which explains the small difference between the RMS error of the efficient filter and the nudged-only ensemble). Beyond a nudging intensity of $K = 0.5$, the filter does worse than the nudged ensemble. The limiting nudging parameter above which the filter is not expected to succeed, which is very small for non equalized weights, is hence shifted as a consequence of weight equalization, to a higher threshold value, which guarantees success of the filter for stronger nudging.

5.4.4 Conclusion

In section 5.4 we tried to understand under which circumstances the efficient particle filter works better compared to a nudged ensemble that does not undergo resampling. The filter works for certain constellations of parameters, meaning that it outperforms a nudged-only ensemble of particles in terms of mean RMS error, while it does not for others. The filter stops working well when a certain threshold value of nudging parameter K is attained.

We found out that the standard deviations of the density exponents play a significant role in the selection mechanism of the particles, just before resampling. There is a difference whether only one, or more dynamical variables are assimilated. If only one variable is assimilated, the standard deviation of the observational term in the weight exponent has to be statistically dominant over the exponent of the p/q-fraction in order to guarantee a selection of better situated particles with respect to observations. If more than one variable is assimilated, other more dominant contributions than the observational term influence the selection mechanism, such as a very large contribution of the p density of certain variables, which a priori discriminates other contributions to the variability of the weights. In this case, knowledge about the correlations between dynamical variables might be helpful.

We computed an analytical approximation to the standard deviation of these exponent terms, and we showed that the value of nudging K beyond which the filter stops working well can be estimated by an analytic formula in the case where weights are not equalized.

Finally, we investigated the equalization of weights numerically. This investigation revealed that the numerical values of the standard deviations are significantly modified by the equalization process, to the favour of the observational term which increases and to the detriment of term $2-3$ which decreases. Hence, particles which without weight equalization would not have had a dominant standard deviation over term $2-3$ now keep this qualifying attribute for larger K-values than they would have had without weight equalization, and the filter succeeds for stronger nudging, as well as it is non degenerate. The numerical numbers of the weight equalization confirm the results shown in the former sections (e.g. Figure 5.32).

Chapter 6

Conclusion

We tested the efficient particle filter [33] on two idealized models that resemble convective-scale dynamics. The purpose of this work was to investigate the applicability of new methods in nonlinear and non-Gaussian filtering, with the long-term aim of improving the forecasting of convective scale weather phenomena such as severe storms and precipitation events.

The efficient particle filter tested on the stochastic cloud model showed to perform only under certain restrictive conditions of parameters, such as a specific range of nudging intensity, cloud half life and observation density. Only for specific parameters, the efficient filter is able to outperform the SIR-filter. For the stochastic cloud model, we computed the probability density of the model evolution. This density function of the stochastic model error reveals, since it is different from the observation density, a 'competition' between the evolution density and the observation density in the weight formation. This enabled us to show how the selection mechanism of resampling works in combination with nudging the state vectors towards observations, both inherent to the efficient filter. Precisely this 'extreme' testbed model, as it unfolds a purely stochastic and totally unpredictable behaviour just as intermittent formations of cloud clusters do, shows its limits for the efficient particle filter. It is discrete and therefore not compatible with an essential ingredient of the assimilation method [33] which is nudging, designed to be gradual in intensity in order to prevent spurious gravity waves in a real-world simulation.

Testing the efficient particle filter on the modified shallow water model leads to more realistic test situations, and to more success. Unlike in the former model, whose simplicity and interesting function mechanism made us leave out the procedure of equalizing weights in order to avoid filter degeneracy, we here implemented the equalization procedure, as well as Gaussian model and observation errors as in [33]. Since gradual nudging is now allowed, it was interesting to see how strongly one can nudge before the filter produces increasing errors. For indeed, the filter works, in terms of Root Mean Square error compared to an ensemble that is nudged but not resampled, until a certain threshold of nudging parameter is attained, and as for the stochastic cloud model, only under certain circumstances.

We showed numerically that there is a quantity that gives information about the success of the filter, the standard deviation of the different densities in the weight exponent. We further derived an analytical estimate for these quantities which showed to be roughly in accord with numerical values, and which also indicate a maximum nudging parameter above which the filter is not expected to succeed anymore. Hence, the weight exponents give information about the potential success of resampling, on which particle filtering is based. We also showed that it matters how nudging is performed, and in particular that a diagonal nudging matrix led to better results, for the assimilation of all dynamical values, of the filter. And finally, these experiments showed that the less information is given in terms of observations and of assimilated variables, the more the filter is guaranteed to 'succeed' compared to a simple analogueous method of 'only-nudging' the ensemble of particles.

All in all considered, the efficient particle filter seems to be a method which, as it essentially is ruled by the weights, is very sensitive to the observation and model error densities and hence, to the parameters that are involved in the weights, such as nudging, model and observation error covariances, length of assimilation cycles. We presume it is also sensitive to physical parameters like viscosity. In the future, it might be interesting to test the particle filter on even more realistic test model, and to investigate whether it is possible, based on the knowledge of the error densities, to predict a priori the range of nudging parameters under which the filter can be expected to work. But prior to such tests, further comparisons with other data assimilation methods as EnKF and LETKF seem appropriate, as well as studying whether nonlinear flitering methods are really required. For this, measuring the non-Gaussianity as suggested in [7] and [51], as well as exploring other methods of measuring the success of the filter using rank histograms [33] or scatter plots [11] give hope for more understanding of the matter.

Figure 6.1: Approaching cloud front on the sea (taken from http://qwallpapers.net).

Appendix

Main Numerical implementation steps of the equivalent weight particle filter

- For each integration time step $t \in [0, L-1]$ of an assimilation cycle (e.g. $L = 100$),

- Integrate the model state according to the model equation
- For non-zero nudging
    1. if SIR = False (no resampling):
        - nudge particles
    2. else:
        - For each particle:
            - if $t \leq L-2$:
                - nudge particles
                - build log p and log q of the densities p and q, and sum them up to time $L-2$
                - if $t = L-2$: store log p, log q at time $L-2$
            - if $t = L-1$:
                - compute the nudging parameter α to maximize the weight of each particle
                - nudge the particles with α
                - compute the last time step contribution of log p and log q, and store $\log p(L-1)$, $\log q(L-1)$
                - compute the maximized log of the weights log w(α)
        - end loop over particles
        - if $t = L-1$:
            - sort the weight maxima log w
            - determine the target weight
            - compute the new parameter α_{new}
            - generate last time step perturbation $\boldsymbol{\beta}$

    - ∗ nudge *chosen particles* (i.e. those that can reach the target weight) with α_{new}
    - ∗ Re-compute the last time step contribution of log p and log q
    - ∗ Re-compute the observation, based on α_{new}, at last time step
    - ∗ Re-compute the (log) weights, which are now (nearly all) equalized

- For zero nudging: SIR procedure
  - evolve particles without nudging
  - compute the observation density and the weights
- Resample the particles, and replace the indices of particles and weights by the new index list of resampled particles

- Start a new assimilation cycle.

Bibliography

[1] Ades M., Data Assimilation In Highly Nonlinear Systems, Thesis, 2013

[2] Ades M. and Van Leeuwen P.J., The equivalent-weights particle filter in a high-dimensional system, Quarterly Journal of the Royal Meteorological Society, May 2014

[3] Arulampalam S., Maskell S. et al., A Tutorial on Particle Filters for Online Non-linear/Non-Gaussian Bayesian Tracking, IEEE Transactions on Signal processing (Vol. 50, N2), Feb 2002

[4] Bain A., Crisan D., Fundamentals of Stochastic Filtering, Springer, 2009

[5] Baldauf M. et al., Operational Convective Scale Numerical Weather Prediction with the COSMO Model: Description and Sensitivities, American Meteorological Society, Feb 2011

[6] Bengtsson, T., P. Bickel, and B. Li: Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. IMS Collections, 2, (316 - 334), 2008.

[7] Bocquet M., Beyond Gaussian Statistical Modeling in Geophysical Data Assimilation, Monthly Weather Review, Aug 2010

[8] Bocquet M. et al., Towards the operational estimation of a radiological plume using data assimilation after a radiological accidental atmospheric release, Atm. Environment 45, 17 (2944 - 2955), 2011

[9] Burgers G., van Leeuwen P.J., Evensen G., Analysis scheme in the ensemble Kalman filter, Monthly Weather Review 126 (1719 - 1724), 1998.

[10] Cohn S.E., An Introduction to Estimation Theory, Journal of Meteorological Society of Japan, Vol.75, No.1B, 1997

[11] Cosme E. et al., A deterministic fully non Gaussian analysis scheme for ensemble filters: The Multivariate Rank Histogram Filter, presentation at Annual Meeting of Meteo France, 2012 (http://www.meteo.fr/cic/meetings/2012/ensemble.conference/presentations/session01/4.pdf)

[12] Craig G.C. and Cohen B.G., Fluctuations in an Equilibrium Convective Ensemble. Part I: Theoretical Formulation, 1996 Journal of Atm. Sciences, Vol. 63, 2006

[13] Daley R, Atmospheric Data Analysis, Cambridge University Press, 1991

[14] Doucet A., Godsill S., Andrieu C., On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing, 10 (197-208), 2000

[15] Doucet A., De Freitas N., Gordon N., Sequential Monte Carlo Methods in Practice, Springer, 2001

[16] Evensen G.,The Ensemble Kalman Filter: theoretical formulation and practical implementation, Ocean Dynamics, (53: 343367), 2003

[17] Evenson G., Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys.Res. 99, (10143-10162), 1994

[18] Evenson G., Data Assimilation: The Ensemble Kalman filter, Springer, 2006

[19] Foata D., Fuchs A., Calcul des Probabilités, ed. Dunod, 2003

[20] Gordon N.J., Salmond D.J, Smith A.F.M., Novel Approach to nonlinear/non-Gaussian Bayesian state estimation, IEE Proceedings, Part F, 140(2):107-113, 1993

[21] Grimsdell A. W. et al., Model Study of Waves Generated by Convection with Direct Validation via Satellite, J.Atm. Sciences (Volume 67, 16171631), May 2010

[22] Hamill T.M., Whitaker J.S., Snyder C., Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter, Mon. Wea. Rev. 129 (27762790), 2001

[23] Houtekamer, P. L., L. Lefaivre, J. Derome, H. Ritchie, and H. L. Mitchell, A system simulation approach to ensemble prediction. Mon. Wea. Rev. 124 (1225-1242), 1996.

[24] Houze R.A., Cloud Dynamics, Academic Press Inc., 1993

[25] Hunt B. R., Kostelich E.J. et al., Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, Elsevier, Physica D 230 (112-126), 2007

[26] Kalnay E., Atmospheric Modeling - Data Assimilation and Predictability, Cambridge University Press (1st ed.), 2002

[27] Karlin S., Taylor H.M., A first Course in Stochastic Processes, Academic Press, 1975

[28] Lahoz W., Khattatov B., Menard R., Data Assimilation, Springer, 2010

[29] Landau L.D. and Lifshitz E.M., Fluid Mechanics, Course of Theoretical Physics - Volume 6 (2nd Ed.), Pergamon Press, 1987

[30] Lane T.P., Characterization of Momentum Transport Associated with Organized Moist Convection and Gravity Waves, J. Atm. Sciences (67, 32083225), Oct 2010

[31] Lawson W., Hansen J.A., Implications of Stochastic and Deterministic Filters as Ensemble-Based Data Assimilation Methods in Varying Regimes of Error Growth, Monthly Weather Review (Vol. 132), 2004

[32] Van Leeuwen, P.J., Review: Particle Filtering in Geophysical Systems, American Meteorological Society, June 2009

[33] Van Leeuwen, P.J., Nonlinear data assimilation in geosciences: an extremely efficient particle filter, Royal Meteorological Society (136), 2010

[34] Lorenc A., Analysis methods for numerical weather prediction. Quart. J. Roy. Meteor. Soc. 112, 1177-1194, 1986.

[35] Lorenc A., A global three-dimensional multivariate statistical interpolation scheme, Mon. Wea. Rev. 109 (701721), 1981.

[36] Metropolis N. and Ulam S., The Monte Carlo Method, Journal of the American Statistical Association, Vol. 44, 1949

[37] Kain, John S., The KainFritsch Convective Parameterization: An Update. J. Appl. Meteor., 43, 170181, 2004.

[38] Kalman, R.E., A new approach in Linear Filtering and Prediction Problems, Journal of the ASME - Journal of Basic Engineering, 82 (Series D): 35-45, 1960

[39] Kalman, R.E., New Results in Linear Filtering and Prediction Theory, Journal of Basic Engineering, March 1961

[40] Lange H. and G.C. Craig, The Impact of Data Assimilation Length Scales on Analysis and Prediction of Convective Storms, Monthly Weather Review (submitted), 2014

[41] Parrish DF, Derber JC, The National Meteorological Center's spectral statistical-interpolation analysis system, Monthly Weather Review, 1992

[42] Pham D.T., Stochastic Methods for Sequential Data Assimilation in Strongly Nonlinear Systems, Monthly Weather Review vol 129, May 2000

[43] Piani C., A Numerical Study of Three-Dimensional Gravity Waves Triggered by Deep Tropical Convection and Their Role in the Dynamics of the QBO, J.Atm.Sciences (Vol. 57, NO. 22), Nov 2000

[44] Pires C., Talagrand O., Bocquet M., Modelling non-Gaussianity of background and observational errors by the Maximum Entropy method, Physica D: Nonlinear Phenomena 239, 2010

[45] Plant R.S. and Craig G.C., A Stochastic Parametrization for Deep Convection Based on Equilibrium Statistics, J. Atmos. Sci., 65, 87105

[46] Poincaré H., Calcul des Probabilités, Cours de Physique mathématique, Gauthier-Villars, 1912

[47] Richardson L., Weather prediction by numerical process, Cambridge University Press, 1922

[48] Schraff C., Online introduction to KENDA (www.cosmo-model.org), 2012

[49] Snyder C. et al., Obstacles to High-Dimensional Particle Filtering, Monthly Weather Review, December 2008

[50] Snyder C., Mathematical and Algorithmic Aspects of Atmosphere-Ocean Data Assimilation, Performance bounds for particle filters using the optimal proposal (pp 3421), Mathematisches Forschungsinstitut Oberwolfach, Report No. 58/2012, Dec 2012

[51] Pires C. A., Talagrand O., Bocquet M., Diagnosis and impacts of non-Gaussianity of innovations in data assimilation, Physica D 239, 2010

[52] Talagrand O., Courtier P., Variational assimilation of meteorological observations with the adjoint vorticity equation. I.: Theory Q.J.R. Meteorological Society (113: 1311-1328), 1983.

[53] Talagrand O., Courtier P., Variational assimilation of meteorological observations with the adjoint vorticity equation I: Theory, Quart. J. Roy. Meteo. Soc. 113 (13111328), 1987.

[54] Press H., Teukolsky S. A., Vetterling W.T., Flannery B.P, Numerical Recipes: The Art of Scientific Computing (3rd ed.), Cambridge University Press, 2007

[55] Wiener N., Extrapolation, Interpolation and Smoothing of Stationary Time Series, MIT press, 1949

[56] Whitaker J.S. et al., Ensemble data assimilation without perturbed observations, Monthly Weather Review, 130 (19131924), 2002

[57] Würsch M., Toy models for testing data assimilation methods on the convective scale, 2013, thesis

[58] Würsch M., Craig G.C., The impact of localization and observation averaging for convective-scale data assimilation in a simple stochastic model, Q.J.R. Met. Soc., Vol. 139, (p. 515523), January 2013

[59] Würsch M., Craig G.C., A simple dynamical model of cumulus convection for data assimilation research, Meteorolog. Zeitschrift, 2013

[60] Zhang F. et. al., Cloud-Resolving Hurricane Initialization and Prediction through Assimilation of Doppler Radar Observations with an Ensemble Kalman Filter, Monthly Weather Review, (Vol. 137, p2105-2125), July 2009

Acknowledgements

First of all, I would like to thank Prof. George Craig for supervising this thesis. I am very indebted to him for his guidance, patience, and encouragement during these past years. I learned a lot from his enthusiasm in science, which was very inspiring and helpful for me.

I sincerely thank Tijana Janjic, who took care with an endless patience of the co-supervision of my thesis and who spent a lot of time helping me with correcting and proofreading this text.

In addition, I want to express my special gratitude to Prof. Wolf-Christian Müller for his encouragement and support in the last years, and for accepting to be the second reviewer of this thesis.

Many thanks to Martin Weissmann for his advice and explanations in meteorology, and also for help with proofreading. I further thank Christian Keil for his friendly assistance. I address special thanks to Emmanuel Cosme, especially for his precious time and explanations during a visit in Grenoble. I also thank Peter Jan van Leeuwen for his advice and explanations, which helped me to get more insight into the equivalent weight particle filter, during a visit to Reading. I am grateful to Olivier Talagrand, to Marc Bocquet, to Emmanuel Cosme and to Eric Blayo for their kind support and for having given me the opportunity to take part at the summer school on data assimilation methods in Les Houches in 2012. I warmly thank Anna Shlyaeva for her interesting explanations, and discussions during some meetings. I learned a lot from her. I address many thanks to Roland Potthast, for his support and for interesting discussions on mathematics during meetings at DWD or at the LMU. I thank my colleagues of the Meteorological Institute and of the HErZ branch of DWD, and the German Weather Service for giving me the possibility to take part in this research network. I also want to thank my husband Sergei for his support and encouragement during my PhD work. Without him, it would have been much more difficult to finish this project.

This study was carried out in the Hans-Ertel-Centre for Weather Research (HErZ). The HErz research network of universities, research institutes, and the Deutscher Wetterdienst is funded by the BMVI (Federal Ministry of Transport, Building and Urban Development).